HILBERT PHASE METHODS FOR GLOTTAL ACTIVITY DETECTION

BY

ROGER D. SERWY

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Doctoral Committee:

Professor Mark Hasegawa-Johnson, Chair
Professor Stephen Levinson
Associate Professor Michael Oelze
Professor Pramod Viswanath

ABSTRACT

The 2π discontinuities found in the wrapped Hilbert phase of the bandpass-filtered analytic DEGG signal provide accurate candidate locations of glottal closure instances (GCIs). Pruning these GCI candidates with an automatically determined amplitude threshold, found by iteratively removing from the full signal the inlier samples within a fraction of its standard deviation until converged, yields a 99.6% accurate detection system with a false alarm rate of 0.17%. This simpler algorithm, named Glottal Activity Detector For Laryngeal Input (GADFLI), outperforms the state-of-the-art SIGMA algorithm for GCI detection, which has a 94.2% detection rate, but a 5.46% false alarm rate. Performance metrics were computed over the entire APLAWD database, using an extensive, hand-verified markings database of 10,944 waveforms. A related proposed algorithm, QuickGCI, also makes use of Hilbert phase discontinuities, and does not require a thresholding post-processing step for GCI selection. Its performance is nearly as good as GADFLI. Both proposed algorithms operate using the electroglottographic signal or acoustic speech signal.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

APLAWD(W)	Archivable Priority List Actual-Word Database (in Wav format)
AUC	Area Under the Curve
DECOM	DEgg Correlation-based method for Open quotient Measurement
DEGG	Differentiated Electroglottograph
DTFT	Discrete Time Fourier Transform
DYPSA	Dynamic Programming Projected Phase-Slope Algorithm
EGG	Electroglottograph
FFT	Fast Fourier Transform
FRI	Finite Rate of Innovation
GADFLI	Glottal Activity Detector For Laryngeal Input
GCI	Glottal Closure Instant
GMM	Gaussian Mixture Model
GOI	Glottal Opening Instant
HPF	High Pass Filter
HQTX	High Quality Time of excitation
HR	Hit Rate
LP	Linear Prediction
LPF	Low Pass Filter
PDA	Pitch Detection Algorithm

ROC	Receiver Operating Characteristics
SEDREAMS	Speech Event Detection using the Residual Excitation And a Mean-based Signal
SIGMA	Singularity in EGG by Multiscale Analysis
SWT	Stationary Wavelet Transform
TXGEN	Time of eXcitation GENerator
VFCA	Vocal Fold Contact Area
YAGA	Yet Another GCI Algorithm
ZFR	Zero Frequency Resonator

CHAPTER 1

INTRODUCTION

Speech processing, speech pathology, vocal training, and speech recognition systems may require the knowledge of glottal closure instants (GCIs) to augment its processing. Such augmentation can lead to better segmentation of speech [1], improve pitch modification for aspiring singers [2], and provide more accurate vocal pathology diagnosis [3].

A GCI occurs when the vocal cords come together rapidly which introduces an acoustic pulse into the vocal tract, and repeats rapidly to create the human voice. This dissertation describes how to identify the GCIs using the wrapped analytic phase angle of the differentiated electroglottograph signal (DEGG). The method works successfully with GCI identification in speech signals, and provides generalization for the zero-frequency-resonator (ZFR) method [4].

When simultaneous EGG and speech waveform recordings are available, the GCI markings from EGG signals can be used as the reference for benchmarking the performance of GCI detection algorithms that operate on the speech waveform only. Therefore, the performance of GCI detection using EGG signals impacts greatly the performance metrics of algorithms that detect GCIs from speech. Many papers have relied on automatic EGG marking for benchmarking speech algorithms: [4], [5], [6].

In order to benchmark the performance of GCI detection from EGG signals, ground truth reference markings needed to be generated and it has resulted in better understanding of the strengths and weaknesses of existing algorithms and for testing new algorithms.

CHAPTER 2

LITERATURE REVIEW

2.1 The Hilbert Transform and Analytic Signal

The Hilbert transform $\mathcal{H}$ of a signal $x(t)$ is linear and defined [7] as:

$$\mathcal{H}(x(t)) = \text{p.v.} \int_{-\infty}^{\infty} \frac{x(\tau)}{\pi(t-\tau)} d\tau \tag{2.1}$$

which takes the Cauchy principle value of the integral (needed because of the singularity at $t = 0$). It may be considered a convolution of $x(t)$ with the kernel

$$h(t) = \frac{1}{\pi t} \tag{2.2}$$

The Fourier transform $\mathcal{F}$ of the Hilbert transform kernel can be expressed as:

$$\mathcal{F}\left(\frac{1}{\pi t}\right) = -j\,\text{sgn}(\omega) \tag{2.3}$$

with

$$\text{sgn}(\omega) = \begin{cases} -1, & \text{if } \omega < 0 \\ 0, & \text{if } \omega = 0 \\ 1, & \text{if } \omega > 0 \end{cases} \tag{2.4}$$

which in total has the effect of preserving the amplitude of all frequency components (except at $\omega = 0$) while introducing a $\pi/2$ phase shift.

The Hilbert transform may be used to derive the analytic signal which is a useful, complex-valued signal whose Fourier transform has no negative frequencies components. Let $x_a(t)$ be the analytic signal for $x(t)$, which can be defined as:

$$x_a(t) = x(t) + jx_h(t) \tag{2.5}$$

where $x_h(t)$ is the Hilbert transform of $x(t)$. Given this complex-valued signal, it can be re-expressed in polar form as:

$$x_a(t) = M(t) \exp\left(j\phi(t)\right) \tag{2.6}$$

where

$$M(t) = |x_a(t)| = \sqrt{x^2(t) + x_h^2(t)} \tag{2.7}$$

$$\phi(t) = \arg\left(x_a(t)\right) \tag{2.8}$$

The value $M(t)$ is referred to as the envelope of the analytic signal. The arg function computes the complex argument, or phase angle, of the complex value, and should not to be confused with the computation of arctangent,[1] which is correct only when $x(t) > 0$.

The principle value of $\arg(a + jb)$, which spans $[-\pi, \pi)$ can be computed using a two-argument function $\arctan_2$ below:

$$\arctan_2(b, a) = \begin{cases} \arctan(\frac{b}{a}) & \text{if } a > 0 \\ \arctan(\frac{b}{a}) + \pi & \text{if } a < 0 \text{ and } b \geq 0 \\ \arctan(\frac{b}{a}) - \pi & \text{if } a < 0 \text{ and } b < 0 \\ +\frac{\pi}{2} & \text{if } a = 0 \text{ and } b > 0 \\ -\frac{\pi}{2} & \text{if } a = 0 \text{ and } b < 0 \\ \text{undefined} & \text{if } a = 0 \text{ and } b = 0 \end{cases} \tag{2.9}$$

Many numerical implementations, e.g. Numpy and MATLAB, replace the undefined case with zero.

2.1.1 Phase Unwrapping

In many applications, the phase angle of a complex signal may span many branches which manifests as 2π discontinuities due to the principle value being computed. These discontinuities can be removed using a numerical procedure known as phase unwrapping. Several implementations exist, for

[1] Many papers make this mistake when defining the phase angle, e.g. [8]. Take care not to propagate this error.

example, [9] used an adaptive, integration scheme on the phase derivative, [10] used cubic splines, and [11] used root-finding methods for polynomials. A great deal of effort has been put toward developing robust phase-unwrapping algorithms, many of which are used for Hilbert phase analysis.

2.1.2 Discrete Hilbert Transform

The discrete Hilbert transform, when considered from the point of view of band-limited sampling theory, can be expressed rather simply. Consider a digital impulse which has an analog representation of a sinc function as shown in Fig. 2.1, where

$$\text{sinc}(t) = \frac{\sin(t)}{t} \tag{2.10}$$

The solid dots represent the time instances of periodic sampling, every $t = n\pi$. When sampled, the sinc has its peak value of unity at $t = 0$, while every other sample occurs when the sinc evaluates to zero.

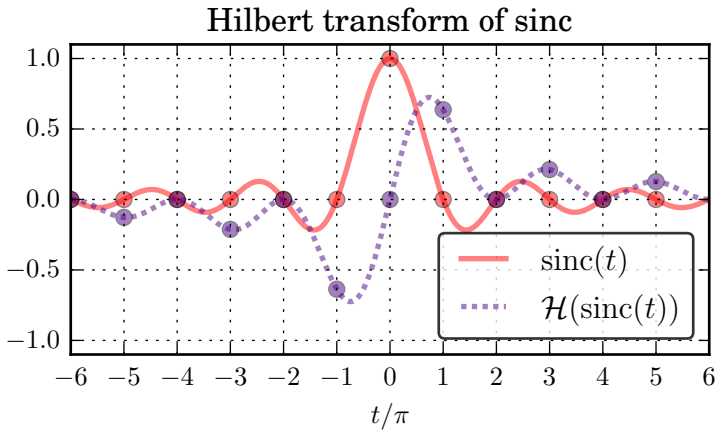


Figure 2.1: Rendering of sinc(t) and its Hilbert transform. The circles represent the discrete-time samples of each continuous-time signal.

The Hilbert transform of sinc(t) is:

$$\mathcal{H}\left(\frac{\sin(t)}{t}\right) = \frac{1-\cos(t)}{t} \tag{2.11}$$

[12], and is shown as the purple, dashed curve in Fig. 2.1. Periodically sampling the Hilbert-transformed sinc gives the discrete Hilbert transform

kernel:

$$h_d[n] = \begin{cases} \frac{1-cos(\pi n)}{\pi n}, & \text{if } n \neq 0 \\ 0, & \text{if } n = 0 \end{cases} \tag{2.12}$$

$$= \begin{cases} \frac{2}{\pi}\frac{\sin^2(\pi n/2)}{n}, & \text{if } n \neq 0 \\ 0, & \text{if } n = 0 \end{cases} \tag{2.13}$$

$$= \begin{cases} \frac{2}{\pi n} & \text{if } n = \text{odd} \\ 0, & \text{if } n = \text{even} \end{cases} \tag{2.14}$$

$$\tag{2.15}$$

which are some of the several ways the kernel has been expressed in the literature, although derived by different methods. Proakis and Manolakis [13] showed the derivation through integrals of the Fourier expression, [14] used transform tables for a time-domain expression, and [15] used a cotangent relationship for transformed periodic signals.

2.1.3 Computing the Analytic Signal

A numerically expedient method for computing the Hilbert transform computes the analytic signal instead, first by computing the FFT of the N-point signal, applying a weighting function $a[m]$ (shown later in Eq. 2.16) to the frequency components, and then applying an inverse FFT. The final result gives the complex-valued analytic signal, whose real component is the original signal and the imaginary component is its discrete Hilbert transform [16]. This approach is used in common scientific processing libraries, such as MATLAB and the Python SciPy package.

The weighting function $a[m]$ used in [16] has the form for N even:

$$a[m] = \begin{cases} 1 & \text{for } m = 0 \\ 2 & \text{for } 1 \leq m \leq \frac{N}{2} - 1 \\ 1 & \text{for } m = N/2 \\ 0 & \text{for } N/2 + 1 \leq m \leq N - 1 \end{cases} \tag{2.16}$$

Elfataoui and Mirchandani [17] showed a corner-case deficiency with this

weighting function for signals with significant energy at $m = 0$ and $m = N/2$, e.g. computing the analytic signal for [1, 2, 1, 2][2] would return itself with no imaginary component. Their proposed solution required adding imaginary values to the $m = 0$ and $m = N/2$ terms which introduced an additional zero in the negative frequencies of the DTFT.

2.1.4 Usages of the Discrete Hilbert Transforms

Gold, et al. [18] described the implementation of discrete-time Hilbert transformers, which were the digital counterpart to the analog systems commonly used in radio for envelope detection and demodulation. The literature pertaining to radio communications continues to make use of the Hilbert transform and can be explained further in some communication textbooks, e.g. [19].

2.1.5 Relation to Other Operators

The Hilbert transform, like the derivative operator $j\omega$, introduces a $\pi/2$ phase shift, but in different directions. Unlike the Hilbert transform, the derivative operator has the effect of scaling frequency components proportional to the frequency.

From a signal processing perspective, the Hilbert transform is almost an all-pass filter. If the original signal $f(t)$ has no DC offset, e.g. $\int_{-\infty}^{\infty} f(t)dt = 0$, then its Hilbert transform preserves all the information of the original signal.

2.2 The Electroglottograph

Electroglottography (EGG) provides a non-invasive measurement of the larynx, which houses the vocal cords used for speech voicing. Since its introduction by [20], it has been used extensively in speech research and clinically for identifying pathologies [21].

The device operates by measuring the electrical impedance across the larynx with two electrodes, using a carrier frequency between 1-5 MHz [22].

[2]The sequence can be longer and have the same effect.

Carrier frequencies below 1 MHz have weak EGG signals [23]. Changes to the larynx impedance amplitude-modulates the carrier, which is then demodulated and provided as a baseband signal. Variations of the EGG exist, such as the multichannel configuration in [24], which can be used to better place the electrodes on the larynx.

The conductance measured by the EGG relates to the contact area of the vocal cords [25], [26]. Increasing the vocal cord contact area, sometimes referred to as vocal fold contact area (VFCA), increases conductance across the larynx.

The EGG is not a perfect means of identifying glottal behavior. It measures the impedance across the larynx, which has more that just vocal vibrations affecting the measured value. For example, [27] showed that mucus strands can complicate the EGG signal, due to the strand maintaining a conductive path across the folds despite the VFCA decreasing. Motion of the neck and electrodes shifting affect the quality of the EGG signal as well.

2.2.1 Interpreting EGG Signals

Over a full glottal cycle, the vocal folds open and then close. Figure 2.2 shows a cartoon depiction of the vocal folds, as well as its corresponding part along the EGG signal, reprinted from [28]. The differentiated EGG, usually referred to as the DEGG, applies a first-difference to the EGG. The glottal cycle starts with a closed glottis (1), which then opens (2-5), and then closes again (6-8). The sudden closure of the glottis is a glottal closure instant (GCI) and occurs at step 6. This corresponds to a large negative pulse in the DEGG. The portion where the glottis starts opening rapidly at step 4 is a glottal opening instant (GOI). The GOI amplitude tends to be much smaller than the GCI amplitude for real EGG signals.

The polarity of the EGG signal affects its interpretation. The EGG shown in Fig. 2.2 has increasing conductance downward, or equivalently higher resistance upward. The DEGG signal then has its GCI spikes occurring downward. Some papers render the EGG signal inverted, e.g. [29], which becomes immediately apparent when inspecting the DEGG and seeings its spikes occur upward. Figure 2.3 shows an example of a simultaneous acoustic and EGG recording, and its DEGG. The low-frequency modulation effects are

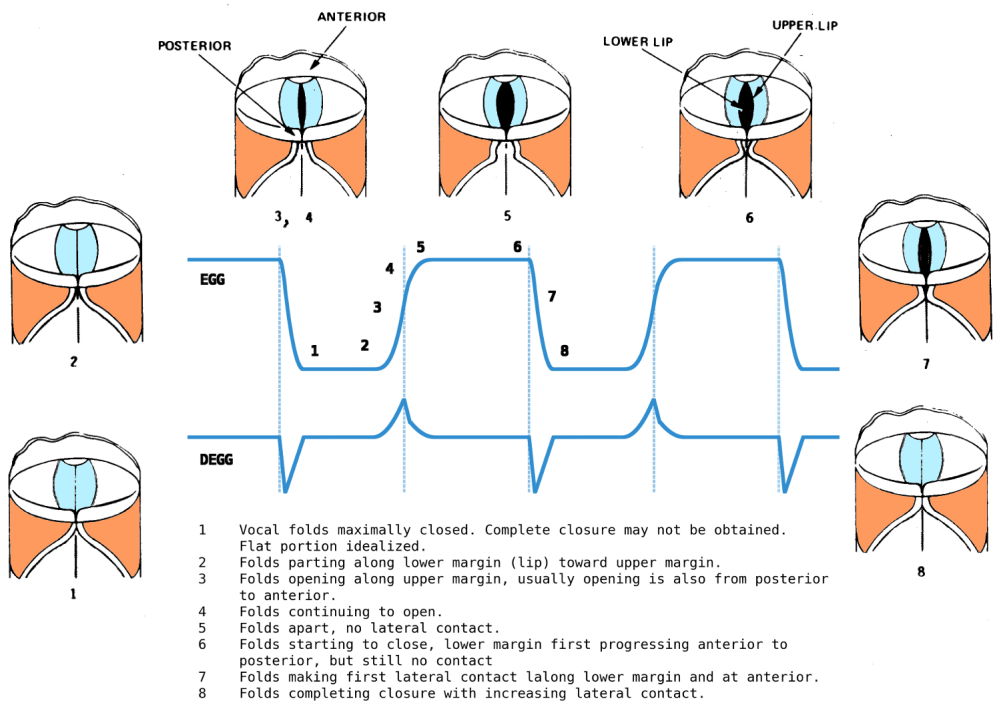


Figure 2.2: The configuration of the vocal folds, and a hypothetical EGG signal, along with its differentiated EGG (DEGG). Adapted from [28].

immediately apparent in the EGG, as the glottal cycle is not zero-mean. The DEGG attenuates low frequencies and amplifies high frequencies, which makes the impulse-like properties of the GCI become readily apparent.

2.3 Using the EGG Signal

The EGG signal itself has found many applications. Smith and Childers [22] used linear predictive (LP) analysis on the EGG signal directly to classify normal and pathological voices due to LP pole placement, which correlated with the higher jitter found in their pathological talkers. Herbst, et al. [30] created "phasegrams", which make use of the analytic EGG signal to plot cycle trajectories and is used to classify different vocal registers.

Herbst, et al. [31] provided a means of visualized the EGG signal cycles using its "wavegram" which period-normalized the length of each glottal cycle in the EGG, and rendered each glottal cycle as grayscale columns. This visualization allows for quick assessment of the VFCA over the duration of speech.

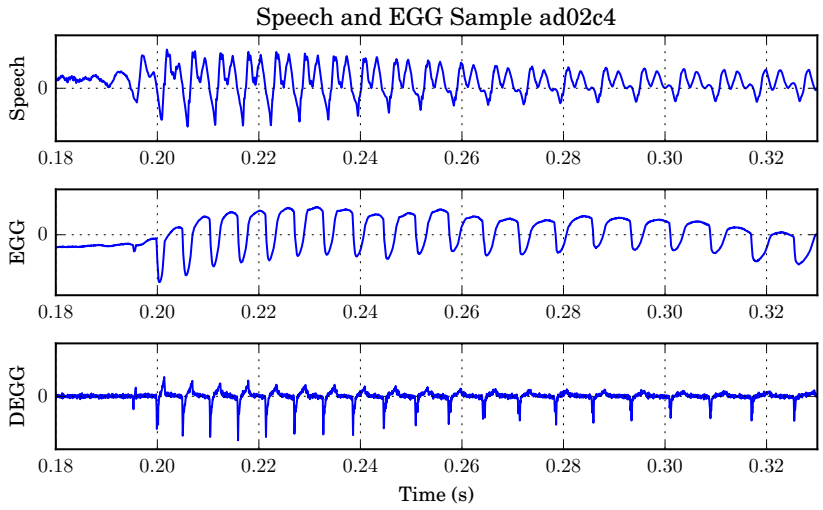


Figure 2.3: A plot of the simulataneous recording of the acoustic and EGG signal of speech. The DEGG is also shown.

Simultaneous capture of the EGG signal along with video of the vocal cords has already validated the utility of the EGG for representing VFCA. Simultaneously capturing the EGG with the acoustic speech signal provides even more opportunities for cross-pollination of ideas. Krishnamurthy and Childers [32] used the EGG to augment the voicing and pitch detection algorithms for acoustic speech processing. Barner [33] made use of an adaptive, non-linear filter to transform speech signals into its DEGG by first training the filter on simultaneous speech and DEGG signals, and then using the filter with speech only. Mooshammer [34] studied the effects vocal effort and stress on the EGG signal statistics and shape.

2.4 EGG-Based GCI Detection

Several algorithms have been proposed for automatically identifying GCIs from EGG signals and acoustic speech signals. Oftentimes, with simultaneous EGG and acoustic recordings, the EGG signal is used to identify reference GCI markings which are then used to benchmark the performance of the acoustic-only algorithms. The knowledge of GCI locations allows for more advanced and accurate speech processing, such as required by pitch-synchronous-overlap-add methods (PSOLA) [2].

GCI detection in acoustic speech is closely related to pitch detection algorithms (PDA). Accurate knowledge of the GCIs provide the pitch period, whereas some speech PDAs do not identify GCIs, such as autocorrelation [35] or cepstral [36] methods.

Hess and Indefrey [37] proposed a pitch detection algorithm[3] which operated directly on the EGG signal, which operated by thresholding and then finding inflection point in the EGG which represents the GCI, and then used spectral methods to iteratively refine the location.

2.4.1 DECOM

Henrich, et al. [29] proposed the DECOM algorithm (DEgg Correlation-based method for Open quotient Measurement) which performed peak-picking on autocorrelated, half-wave-rectified DEGG signals containing the GCIs, and requires the fundamental period be known approximately. Once the refined $F0$ is known, the other half-wave-rectified DEGG containing the GOIs is cross-correlated with the GCI signal, and its peak relates to the open quotient. This method is meant for quasi-steady-state sounds. It can likely be extended to return the GCI locations.

2.4.2 SIGMA

The SIGMA algorithm (Singularity in EGG by Multiscale Analysis) [38], [39] represents the state-of-the-art GCI detection on EGG signals. It operates by identifing possible GCI candidates and then refines this list using a classifier.

SIGMA makes use of a multi-scale product of stationary wavelet transforms, using three iterations. Wavelet transforms operate by applying two wavelet kernels which amount to a high-pass (detail) and low-pass (approximation) filter to a signal. Once applied, the two filtered signals undergo decimation, where every other sample gets discarded. The process can be iterated further on these two output signals, and it is commonly referred to as a multi-level wavelet analysis.

The stationary wavelet transform (SWT), in distinction, does not decimate the signals after filtering but rather interpolates the two kernels by zero-

[3]For the PDP-11.

insertion, where the length of the kernels is doubled, with a zero inserted in between each sample. The next level analysis applies these modified kernels to the previous level's approximation signal. A consequence of the stationary wavelet approach is that the signal under analysis does not change in length.

SIGMA performs a three-level SWT on the EGG signal. The effective transfer function for each of these filters is shown in Fig. 2.4.

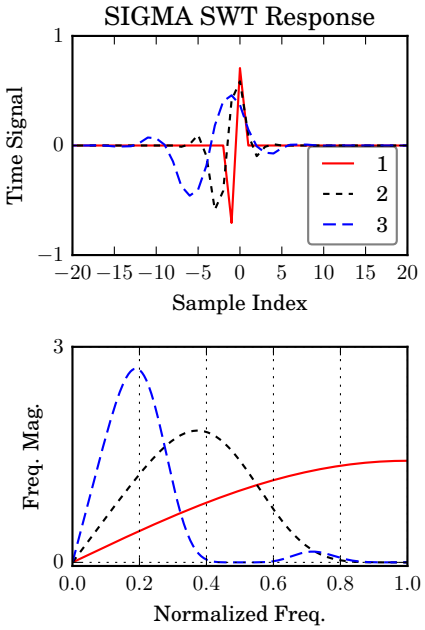


Figure 2.4: Transfer functions of the SWT used by SIGMA. Each level shifts the impulse, while also low-pass filtering.

The detail signals from each level are then multiplied together and then cube-rooted. Since the detail signal is "derivative-like", the output after the first stage gives a DEGG-like signal. The following two stages also produce DEGG-like signals, but with additional low-pass filtering.

The multiplication of the three DEGG-like signals preserves the negative GCI pulses as negative, e.g. $(-1)^3 = -1$. The cube-root operation also preserves the sign of the value, leaving the negative pulses as negative. The cube-root is needed to recover, approximately, the amplitude of the GCI pulses, otherwise the cube of these values will cover a much broader amplitude range than the actual GCI amplitudes cover.

The group delay function can be written as a "center of energy" [40] as:

$$g(n) = \frac{\sum_{m=0}^{M-1} m x_n^2(m)}{\sum_{m=0}^{M-1} x_n^2(m)} \tag{2.17}$$

By its formulation, the group delay function is polarity-agnositic, i.e. processing x or $-x$ gives the same result. In order to avoid marking GOIs as GCIs, SIGMA half-wave rectifies the cube-rooted multi-scale product signal before computing the group delay function.

The negative-to-positive zero-crossings of the group delay function denote the GCI candidates. Each of these candidates have local feature statistics computed (linearity of group delay, amplitude, and area) which are used in a two-class Gaussian Mixture Model (GMM) to separate true GCIs from noise. SIGMA chooses the class whose mean area is largest as the true GCI class.

Some additional post-processing is performed to remove isolated GCI markings (those greater than a 50 Hz period from the nearest marking).

2.4.3 HQTX and TXGEN

Both the HQTX and TXGEN algorithms come from the Speech Filing System software suite [41]. These have been used in prior papers [42], [5], [39] to generate reference GCI markings in EGG signals for evaluating a speech GCI marking algorithm (DYPSA), and the performance of another EGG-GCI algorithm (SIGMA).

The only documentation on the internals of these algorithms exist in their manual pages. TXGEN low-pass-filters the EGG to 3 kHz, applies a first difference, and finds maxima and minima above a threshold. HQTx uses the first difference of the EGG and its instantaneous gradient.

2.4.4 Ensemble Empirical Mode Decomposition

Sharma, et al. [6] proposed a method using Ensemble Empirical Mode Decomposition (EEMD) to identify the location of GCIs from EGG signals. They only make use of the first intrinsic mode function (IMF) found, which strongly resembles the first difference of the DEGG signal, a DDEGG so to speak. The Hilbert envelope of this first IMF is smoothed repeatedly

with a rectangular window moving average filter, and then its local maxima computed to reveal the GCI locations.

The approach performs comparable to SIGMA when tested against the CMU-Arctic database [43] which contains unmarked EGG recordings. The reference GCIs used for benchmarking their approach to SIGMA were generated by another simpler algorithm they wrote, which found the sharp negative peaks in the DEGG.

2.4.5 Finite Rate of Innovation

Amin and Marziliano [44] proposed using the methods from the finite rate of innovation (FRI) to identify the locations of GCIs as well as GOIs from EGG signals. This method assumes that the underlying signal is sparse, and its solution places the impulse at the best location, including fractional samples. The performance was benchmarked against SIGMA, HQTX, and TXGEN, using a hand-labeled subset from the APLAWDW [45] database. In total, 500 EGG waveforms were labeled with reference GCIs.

The FRI approach outperformed SIGMA 99.3% to 97.8%, and also exceeded two other methods. Its voicing activity detector, taken from the STRAIGHT vocoder [46], provided F0 estimates which were then used to constrain the search window for GCIs. This additional constraint reduced the false alarm rate for FRI (0.15%) relative to SIGMA (2.13%).

2.5 Speech-Based GCI Detection

While the literature has many implementations of pitch detection algorithms, there are fewer algorithms that focus on identifying the GCI, sometimes also called an "epoch", in the voiced speech signal.

2.5.1 Zero-Frequency Resonator

Murty and Yegnanarayana [4] proposed a numerically simple (and technically unstable) method for identifying GCIs from the speech signal. The method performed a finite difference on the speech samples, quadruple-applies a 0-Hz resonator filter, which is a first-order IIR filter with a pole at $z^{-1} = 1$, and

then subtracts the local mean (spanning two pitch periods) of the signal. The negative-to-positive zero-crossings of this signal represent the GCIs of the speech signal. For stability reasons, the pole was placed slightly interior to the unit circle. For relatively clean-speech signals, the method gives good results.

An FIR implementation of the ZFR [47] shows a doubled detrending filter, used to help further stabilize the numerics of the method.

2.5.2 DYPSA

Kounoudes, et al. [42] proposed the DYPSA algorithm which uses dynamic programming to find the optimal decision of GCI events. It makes use of the group delay function (see Eq. 2.17) on the LP residual, and then uses the zero-crossings to mark GCI candidates. It dynamic program incorporates various cost functions into its decision: pitch deviation, amplitude consistency, waveform similarity, and phase-slope deviation.

Naylor, et al. [5] benchmarked the performance of DYPSA by using the EGG GCI markings from HQTX on the APLAWDW [45] database.

2.5.3 YAGA

The YAGA algorithm from [48] shares the same front-end approach of SIGMA by using a the group delay function on the multiscale product to identify GCI candidates, and the same dynamic programming back-end approach to classify GCIs with DYPSA. With these enhancements, YAGA outperforms DYPSA, however, its optional voicing detection feature can compromise some performance.

2.5.4 SEDREAMS

Drugman and Thierry [49] proposed the yet-to-be-named SEDREAMS algorithm (later named in [50]). The algorithm operates on a bandpass-filtered version of the speech which resembles a sinusoid. Different parts of the signals' cycle are used, namely the trough of the sinusoid to 1/4 period further,

to limit the samples considered for finding a peak in the LP residual of the speech.

SEDREAMS outperforms DYPSA, YAGA, and the ZFR by almost all metrics given in [50], having a higher hit rate and smaller variance on its timing accuracy.

CHAPTER 3

PROPOSED ALGORITHMS

3.1 Hilbert Transform and Wrapped Analytic Phase

The Hilbert transform relates the real and imaginary components of the analytic signal $x_a(t)$:

$$x_a(t) = x(t) + jx_h(t) \tag{3.1}$$

where $x(t)$ is the real-valued signal and $x_h(t)$ is its Hilbert transform.

Figure 3.1 shows some properties of the analytic signal for $x(t) = \cos(\omega t)$, shown as the solid black line. Its Hilbert transform $x_h(t) = \sin(\omega t)$, shown in the dashed line, demonstrates the $\pi/2$ phase shift while preserving its amplitude. The dotted line shows the instantaneous phase angle.[1]

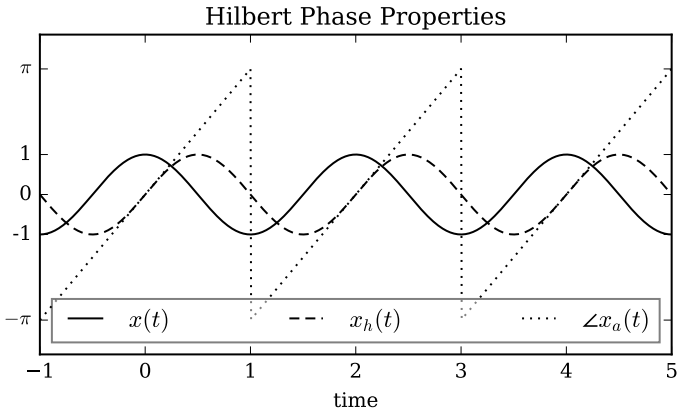


Figure 3.1: Hilbert transform properties for a cosine input signal $x(t) = \cos(\omega t)$, its analytic signal $x_h(t) = \sin(\omega t)$, and its wrapped phase angle $\angle x_a(t)$. The 2π discontinuities mark the beginning of each cycle.

[1]This phase angle can span the full range of $[-\pi, \pi)$. Computing it using the arctangent function will cover only half this region, due to sign ambiguity, i.e. -1/1 = 1/-1 and 1/1 = -1/-1. The implementation of arctan2 exists in many languages, such as C [51], Python, and MATLAB.

The phase angle from Eq. 3.1 would then be:

$$\phi(t) = \arctan_2(x_h(t), x(t)) \tag{3.2}$$

The 2π discontinuities in the phase occur when $x_h(t)$ has a positive-to-negative zero-crossing while $x(t) < 0$.

The time-derivative of $\phi(t)$ gives the instantaneous frequency of $x(t)$ at all times other than these discontinuities. Most Hilbert transform analysis methods remove these discontinuities, but this thesis proposes to preserve these discontinuities, rather than remove them. The instantaneous frequency would then have a 2π impulse occurring every $t = \pi(2n+1)/\omega$ for all n. These impulses may be considered the starting time of a cycle and are easy to detect numerically.

The location of these 2π discontinuities can be shifted to different parts of the cycle by rotating the analytic signal by a phase angle θ and then computing the arctangent of the ratio of the imaginary component to the real component:

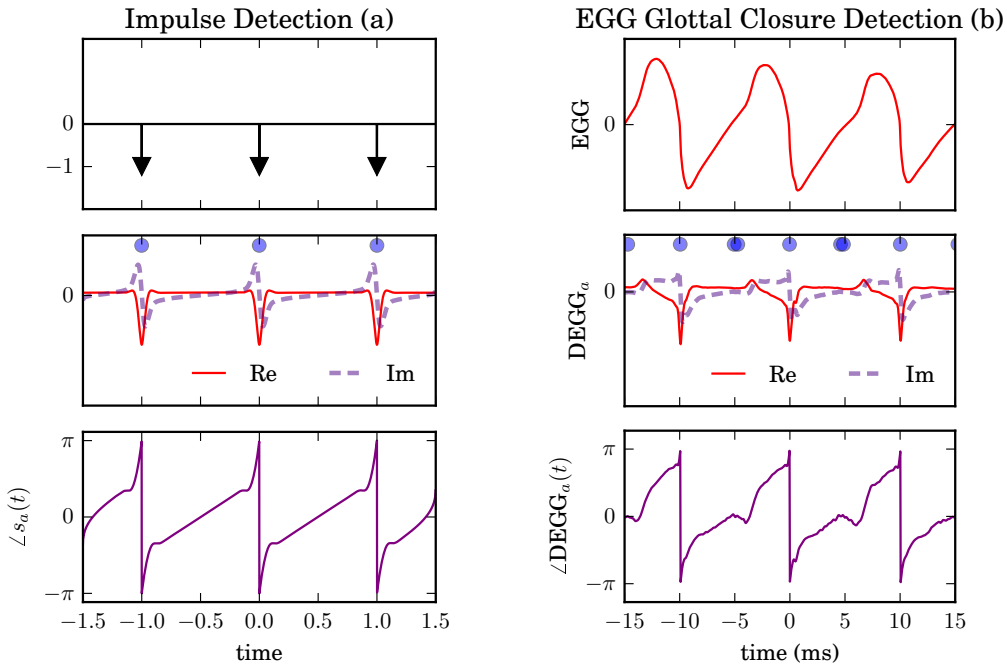
$$\phi_\theta(t) = \arctan_2\left(\mathrm{Im}\{e^{j\theta}x_a\}, \mathrm{Re}\{e^{j\theta}x_a\}\right) \tag{3.3}$$

which generalizes Eq. 3.2. This rotation redefines the starting point of a cycle. This rotation can be useful for tracking different portions of the EGG glottal cycle.

3.1.1 Impulse Detection

Consider a train of negative impulses as shown in the top panel of Fig. 3.2(a). Bandpass-filtering these impulses with a non-causal, symmetric filter preserves the time location of the impulses.[2] Its Hilbert transform is shown as the dashed line, and the time locations of the positive-to-negative zero-crossings are denoted by the dots at the top of the waveform. Its instantaneous phase is shown in the bottom panel. (The line connecting $+\pi$ to $-\pi$ is a rendering artifact of the plotting software, but its presence makes the discontinuities more salient.)

[2]This zero-delay filter can be accomplished by applying the filter forward in time, and then backward in time, which cancels any frequency-dependent phase shifts.



(a) Negative impulse train (top), bandpass-filtered analytic signal (middle), and its instantaneous phase (bottom), where 2π phase discontinuities occur at the impulses.

(b) EGG (top), analytic DEGG (middle) and its phase angle (bottom). The 2π discontinuities align with the glottal closure instances.

Figure 3.2: Example of negative impulse detection using Hilbert phase (left) and applying it to the DEGG signal (right). Time instances of zero-crossings of the imaginary component are marked with blue circles.

Detecting the impulses in the original signal can be accomplished by finding the impulses in the time-derivative of the wrapped analytic phase angle.

Applying this method to the EGG signal reveals the location of the glottal closure instances. Figure 3.2(b) shows the EGG, the analytic DEGG and its phase angle. The 2π phase discontinuities align with the GCIs.

The blue dots above the analytic DEGG signal mark the positive-to-negative zero-crossings of the imaginary signal. These crossings occur more often than the 2π discontinuities, because of the additional requirement for a discontinuity of the real signal being negative at the imaginary zero-crossing instance. This extra requirement acts a kind of filter against spurious zero-crossings, and for speech signals, this zero-crossing filtering becomes a useful enhancement over the ZFR method described later in Section 6.2.

3.1.2 DC Impulse Phase

Consider a negative Gaussian pulse as shown in Figure 3.3, given by

$$p(t) = -e^{-t^2} \tag{3.4}$$

Its Hilbert transform $p_h(t)$ and phase angle $\angle p_a(t)$ are also shown in dashed purple and dotted green, respectively. This pulse has a negative DC component, unlike the bandpass-filtered impulse train shown in Fig. 3.2(a). The pulse's negative DC offset causes its Hilbert phase to settle to $+\pi/2$ for $t \ll 0$ and to $-\pi/2$ for $t \gg 0$. The 2π phase discontinuity occurs at the center of the pulse at $t = 0$, as expected.

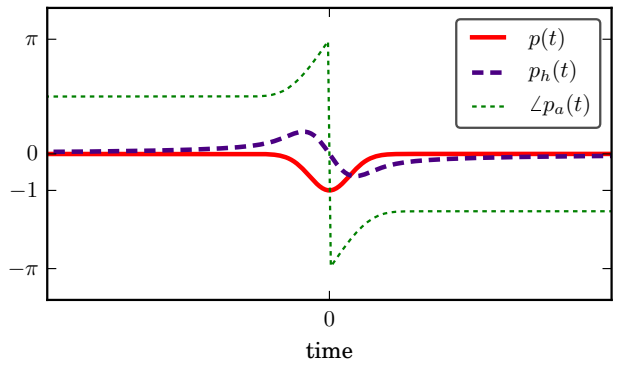


Figure 3.3: A negative Gaussian pulse, its Hilbert transform and phase. Its phase approaches to $\pm\pi/2$ when moving away from the pulse, and is due to the pulse's DC offset.

3.2 Inlier Elimination

The analytic DEGG signal has noise, which especially becomes apparent during periods of no glottal activity. This noise may create many spurious 2π discontinuities in the phase angle. In order to separate these spurious discontinuities from those caused by GCIs, an iterative thresholding procedure has been developed to separate these two classes.

This iterative procedure computes the standard deviation σ_k of the signal and then removes all samples within $\pm\sigma_k \cdot m$, centered around zero, where $0 < m < 1$. The k subscript on σ_k represents the standard-deviation of the

"kept" samples. The process repeats until no more samples can be removed. This final σ_k is then used to amplitude-normalize the signal by dividing the signal by σ_k. More formally, the algorithm proceeds as:

1. Let $\mathcal{B}$ be the set of all samples in the signal.
2. Compute the standard deviation σ_k of $\mathcal{B}$.
3. Keep the larger samples such that $\mathcal{B}^+ = \{n \in \mathcal{B} : |n| \geq m\sigma_k\}$.
4. If $|\mathcal{B}^+| = |\mathcal{B}|$, stop.
5. Use the new subset instead: $\mathcal{B} := \mathcal{B}^+$.
6. GOTO 2.

This algorithm may be thought of as *inlier* elimination, as compared to the outlier elimination approaches which iteratively remove samples exceeding a multiple of the standard deviation, typically three. A theoretical example can be found in Section 3.2.2.

Choosing the amplitude normalization with this method provides improved GCI detection performance over normalizing by the full signal standard deviation which is sensitive to the ratio of the duration of speech to non-speech segments.

3.2.1 Interpretation

The inlier elimination algorithm is merely a heuristic which improves the thresholding performance of GADFLI, as compared with using a normalization based on the waveform's standard deviation. It captures, in an average sense, a gross statistic about the variability of strong parts of the DEGG signal.

This inlier elimination is reminiscent of the center-clipping algorithm used in the autocorrelation pitch detection algorithms in [36]. There, the central portion of the speech signal (a fixed fraction of the peak value over a 30 ms window) was removed before autocorrelation to determine pitch. Here, the inlier elimination is used to determine the size of the central portion of the signal to remove.

3.2.2 Iterative Thresholding Theoretical Example

Consider a tractable example of a sinusoid followed by noise, as shown in Fig. 3.4(a), which can be expressed as:

$$s(t) = \sigma_n \cdot n(t) + \begin{cases} \sqrt{2}\sin(8\pi t), & \text{if } 0 \leq t < 1 \\ 0, & \text{otherwise} \end{cases}$$

where σ_n is the standard deviation of the noise signal and $n(t)$ is a zero-mean Gaussian white noise signal with unity standard deviation. In the case of $\sigma_n = 0$, the first iteration would remove all the silent region samples (and a small set of points near zero in the sinusoidal signal). The set of remaining samples now has an increased standard deviation and every successive step increases it. This algorithm is guaranteed to converge to a non-empty set when $m < 1$.

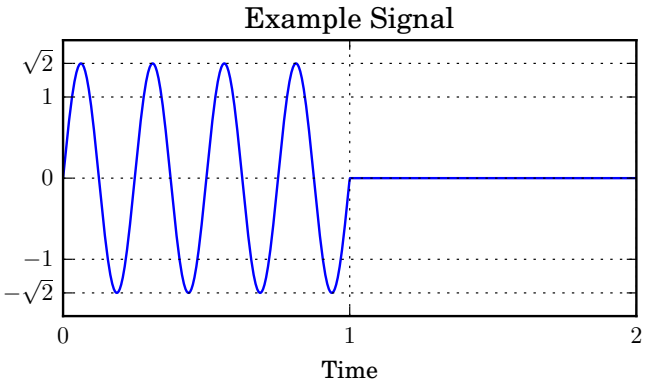
The standard deviation of the kept samples, when the algorithm converges, increases with increasing m, as shown in Fig. 3.4(b), for different noise levels. The standard deviation of the sinusoidal signal itself is unity, and $\sqrt{2}/2$ when spanning both the sinusoid and silence region. When $m = 0$, no samples are removed and thus $\sigma_k = \sqrt{2}/2$. For larger m, the converged value increases. Figure 3.4(b) shows the resulting σ_k for increasing m and for different noise levels, expressed in dB re 1 ($\sigma_n = 10^{x/20}$).

All the curves converge to the no-noise condition near $m = 0.4$. In this example, having a signal-to-noise ratio of 18 dB or better, and choosing $m = 0.3$ will converge to a value that has almost no dependence on the noise level.

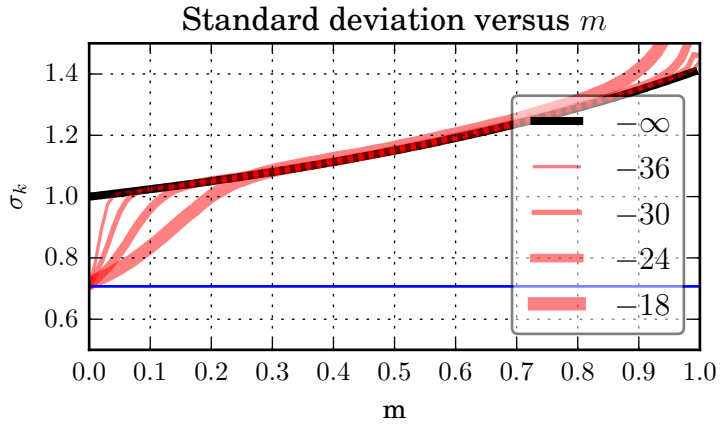
3.3 The GADFLI Algorithm

The GADFLI (Glottal Activity Detector for Laryngeal Input) algorithm operates by combining the detection of 2π discontinuities in the phase of analytic DEGG signal in Section 3.1.1 with the iterative amplitude threshold described in Section 3.2. Figure 3.5 shows its full block diagram.

GADFLI starts with processing the EGG signal with a linear-phase, zero-delay bandpass filter to preserve the timing of the GCI. Its time derivative is taken, analytic signal computed and rotated (optionally). The wrapped



(a) An example signal used for separating the active region from silence.



(b) Standard deviation σ_k of the remaining set of samples as a function of m and additive noise level, expressed in dB re 1.

Figure 3.4: An example signal and its iterated standard deviation of kept samples with increasing levels of noise.

phase angle of the rotated analytic signal is searched for positive-to-negative 2π jumps. The real component has its amplitude normalized by σ_k found by using the inlier elimination algorithm. A threshold τ is applied, using the DEGG amplitude at the 2π discontinuity locations. Empirically, setting $m = 0.25$ for inlier elimination and the amplitude threshold to $\tau = -0.25$ gives good results, which will be explained in more detail in the coming sections (see Section 5.1.3).

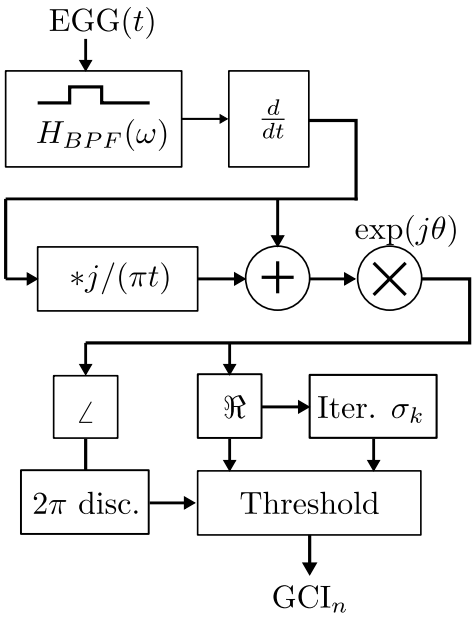


Figure 3.5: Block diagram of GADFLI for GCI detection.

3.3.1 Filter Design and Application

The bandpass filters used in GADFLI are first-order high-pass and low-pass filters. The transfer function for a first-order digital filter can be expressed as:

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}} \tag{3.5}$$

For a given 3 dB analog transition frequency F_c, and sampling rate F_s, the coefficients for the digital high-pass filter and low-pass filters can be computed using the following equations (equivalent to a Butterworth filter):

$$a_1 = -\frac{1 - \tan(\pi F_c/F_s)}{1 + \tan(\pi F_c/F_s)} \tag{3.6}$$

with the numerator coefficients for the low-pass filter expressed as:

$$b_0 = b_1 = \frac{1 + a_1}{2} \tag{3.7}$$

and the high-pass filter expressed as:

$$b_0 = -b_1 = \frac{1 - a_1}{2} \tag{3.8}$$

The two edge frequencies of the bandpass filters are the only filter parameters to configure. This filter design was chosen to avoid the tyranny of tweaking free parameters. The common analog filter prototypes of Butterworth, Chebyshev, and Elliptical, all have the same form in first order and also have the same coefficients when the stop-band and pass-band attenuation values are 3.0103 dB (when applicable).

The impulse response of first-order filters do not have oscillatory behavior found in second- and higher-order filters. These oscillations can be tracked by Hilbert phase during intervals of low signal amplitude and may introduce spurious 2π discontinuities in the analytic phase angle signal.

These filters are applied forward and backward twice to cancel group delay, effectively creating a zero-phase, non-causal filter with a rolloff of 24 dB/octave.

Bandpass filtering of the EGG signal eliminates the low-frequency signal caused by non-glottal activity (swallowing, movement of sensors during speech, etc.), and high-frequency noise (especially present in the APLAWD corpus). For many DEGG signals, low-pass filtering reveals GCIs otherwise occluded by noise when plotted.

3.3.2 2π Discontinuities

Searching for 2π jumps in the discrete-time signal will not yield any results due to the sampling of the signal not aligning with the moment of the jump. A threshold of 1.5π is used instead and yields accurate results. Low-pass filtering the signal reduces spurious jumps caused by noise and smooths the angle signal.

The rotation term on the analytic signal is not used for GCI detection with EGG signals, hence $\theta = 0$. For GOI detection this rotation is $\theta = \pi$ to identify the pulses of opposite polarity. When applying GADFLI to GCI detection in speech signals, there is a rotation of $\theta = -\pi/2$, which is needed to reverse the phase effects of the first-difference in GADFLI, which will be explained partly in Section 6.4.

Figure 3.6 shows a full example of GADFLI, where the glottal closure candidates are chosen by finding jumps greater than 1.5π in the wrapped analytic phase, and then keeping those candidates whose DEGG amplitude

falls below $-\sigma_k/4$ computed for $m = 0.25$. The blue line represents the decision threshold which separates the GCI green circles from the black diamond rejects.

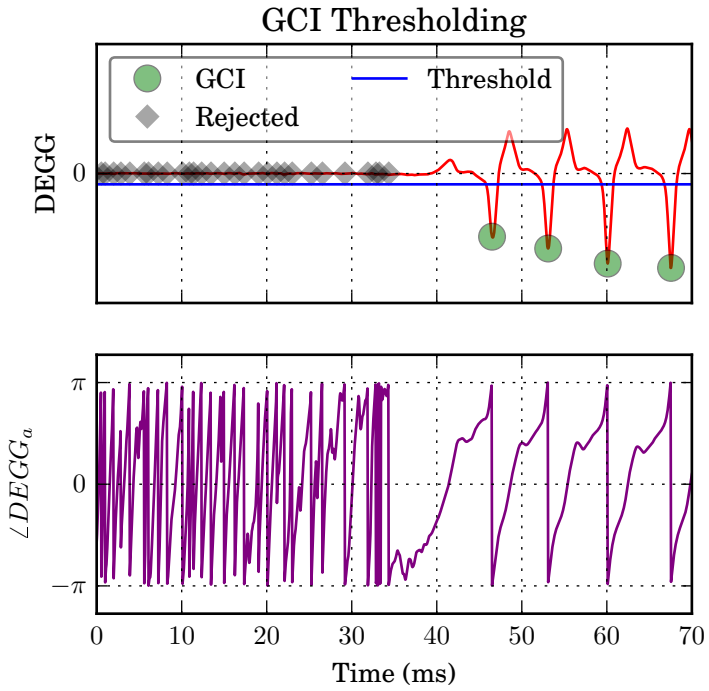


Figure 3.6: The GCI candidates (green and black) from the top panel align with the 2π discontinuities of the lower panel (rendered as vertical lines). An iterative threshold was determined from the DEGG signal (blue line), and the GCI candidates classified into rejected (black diamonds) and not rejected (green circles).

3.3.3 GOI Detection

The Hilbert phase angle of the analytic DEGG signal can be rotated to instead lock onto the GOI instances by setting $\theta = \pi$. Figure 3.7 shows an example of GOI and GCI detection, along with its rotated analytic phase angles.

Since GOIs tend to have a smaller amplitude than GCIs, the sensitivity threshold τ for GADFLI should be adjusted. The resulting GOI locations can be further refined to remove spurious markings by keeping those within a known glottal interval using the detected GCI markings. For some EGG

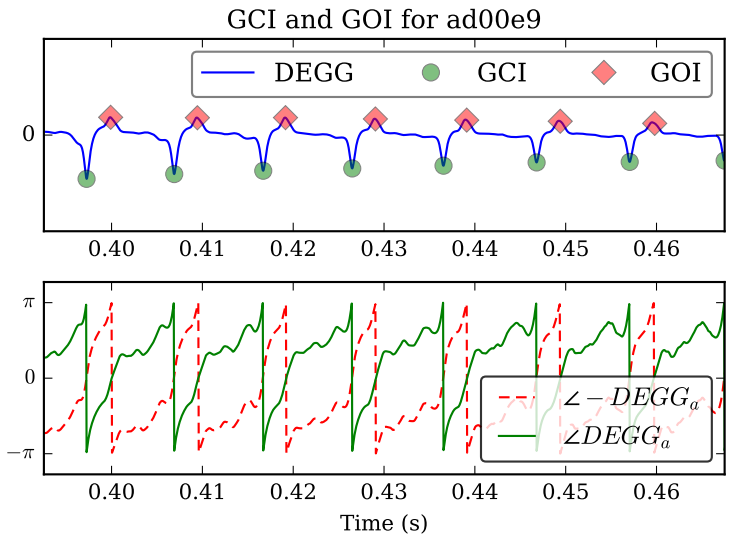


Figure 3.7: Example of GOI detection using GADFLI. The top panel shows an example DEGG with its GOIs (red diamonds) and GCIs (green circles) marked. The bottom panel shows the analytic DEGG phase angle (solid green) and π rotated, which effectively is multiplying the EGG by -1 before processing (dashed red).

waveforms, multiple GOI candidates may exist between two GCIs. Empirically, choosing that last GOI candidate between two GCIs tends to mark the strongest GOI impulse.

In the example from Fig. 3.7, the GOIs occur very closely to the GCI. The open-quotient of this portion of speech nears 75%. If the Hilbert phase were locked onto the fundamental starting at the GCI, the GOI would then be marked near the middle of the cycle. Using a broad-band DEGG signal allows for more precise localization of the GCI and GOI instances as compared to using a narrow bandpass filter centered near the fundamental, like the filtering used by the ZFR method.

A reference database for GOI markings was not prepared, thus no further analysis of GOI performance will be given.

3.4 The QuickGCI Algorithm

The GADFLI algorithm combines the wrapped Hilbert phase approach with an iterative threshold to classify GCIs. Rather than require a classifier as the

last step, the QuickGCI algorithm achieves similar performance to GADFLI with a simpler algorithm, and can find GCIs in acoustic speech and electroglottograph signals.

For relatively clean speech and EGG signals, QuickGCI behaves as a voice activity detector by marking regions with voicing and not marking regions of non-voicing.

3.4.1 Transformation Steps

Formally, the transformative steps for QuickGCI are as follows:

1. Apply a first-order high-pass and first-order low-pass filter (see Section 3.3.1) to the input signal[3] $g(t)$, forward and backward twice in time to preserve GCI locations.

$$x(t) = H_{HPF} * H_{LPF} * g(t) \tag{3.9}$$

2. Compute the analytic signal for $x(t)$ by taking its Hilbert transform and allow for rotation by θ:

$$x_a(t) = [x(t) + jx_h(t)] \exp(j\theta) \tag{3.10}$$

3. Multiply the envelope by the negative imaginary component of the analytic signal.

$$q(t) = |x_a(t)| \cdot \operatorname{Im}[-x_a(t)] \tag{3.11}$$

4. Low-pass filter the signal $q(t)$ to smooth high-frequency self-modulations.

$$r(t) = H_{LPF} * q(t) \tag{3.12}$$

5. Compute the analytic signal of $r(t)$ and find its positive-to-negative 2π phase discontinuities.

$$\phi(t) = \arg(r_a(t)) \tag{3.13}$$

Figure 3.8 gives an example of these transformations on an EGG signal. The top panel shows the bandpass-filtered, analytic EGG signal, along with

[3]The g denotes glottal signal, which applies to EGG and speech.

its enveloped in dotted black. The peaks in the envelope loosely, but not always, correspond to the GCIs. The middle panel shows the analytic, self-modulated EGG signal $r_a(t)$. The solid blue line in the middle panel is the negative, envelope-modulated dashed purple line from the top panel. The bottom panel shows the analytic phase angle of $r_a(t)$.

The 2π discontinuities are the GCIs. Regions of no speech activity have phase near $\pm\pi/2$, or possibly having a negative-to-positive 2π discontinuity which we ignore.

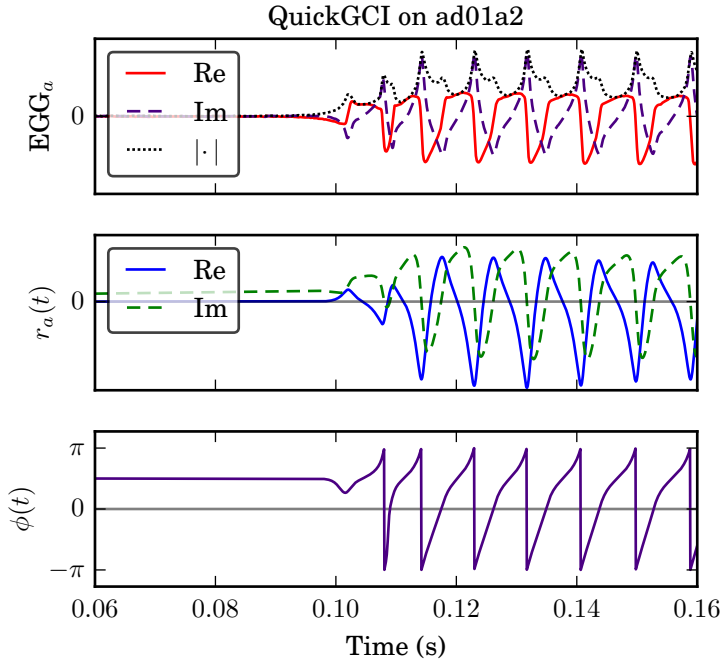


Figure 3.8: An example of QuickGCI applied to ad01a2. The top panel shows the bandpass-filtered EGG signal (Re), along with its Hilbert transform (Im) and its envelope. The middle panel shows the analytic signal of the self-modulated EGG signal. The bottom panel shows its phase angle.

3.4.2 Explanation

If we use the signal transformations of GADFLI from Fig. 3.5 as a conceptual template for explaining QuickGCI, some of the uncertainty over its behavior may subside. For now, let us consider only an EGG signal.

Step 1 applies a bandpass filter to the input signal, just like GADFLI.

In GADFLI, the next step computes the first-difference of the signal, then its analytic signal, and finally applies a rotation. QuickGCI omits the first-difference and only computes a rotated analytic signal in Step 2. We are now lagging by $\pi/2$ phase relative to the GADFLI signal.

Step 3 should be considered in two parts. First it takes the imaginary component of the negative analytic signal. The imaginary component applies a $\pi/2$ phase shift in the opposite direction of the first difference in GADFLI. This is why the negative analytic signal is used – to realign phase. For an EGG signal, the value of $\mathrm{Im}[-x_a(t)]$ resembles a DEGG signal, but its GCI pulses are wider due to the presence of the relatively higher-amplitude lower-frequency components. See the dashed purple line in top panel of Fig. 3.8, and imagine its negative.

The second part of Step 3 multiplies the DEGG-like signal by its envelope. This DEGG-like signal is zero-mean. Its analytic envelope has peaks around the GCIs, and smaller values near the GOIs. When multiplied, the larger values become larger, relative to the local smaller values. Now the signal $q(t)$ is no longer zero-mean; it has a DC offset. This newly introduced DC offset property is critical for the glottal activity detection property of QuickGCI (see Section 3.1.2).

Step 4 applies a low-pass filter to the newly self-modulated signal. Strictly it is not necessary, however it does improve scoring performance.

Step 5 computes the Hilbert phase angle of this DC-offset signal. This DC offset pushes the computed phase angle toward $\pi/2$ in regions where there is low signal. Rather than having the real and imaginary components have comparable magnitudes as with GADFLI which causes the spurious 2π discontinuities in regions of noise, the imaginary component dominates the voiceless regions and pins the phase angle to near $\pm\pi/2$.

Using the envelope is reminiscent of [52] who showed that for high-pass filtered (HPF) speech, the ZFR fails to identify GCIs. By applying the ZFR method to the Hilbert envelope of the high-pass-filtered speech signal, the GCIs could be identified once more. This observation indicates that the Hilbert envelope contains information about the GCI. Many, but not all, of

the local peaks in the envelope from Fig. 3.8 align with the GCIs.[4]

3.4.3 Speech Example

For speech signals, the same GADFLI requirement of rotating the analytic signal $\theta = -\pi/2$ still applies for QuickGCI. Figure 3.9 shows an example of QuickGCI on a speech signal for the phrase "six plus three equals nine." The high-pass and low-pass filters had cut-off frequencies of 50 Hz and 500 Hz, respectively.

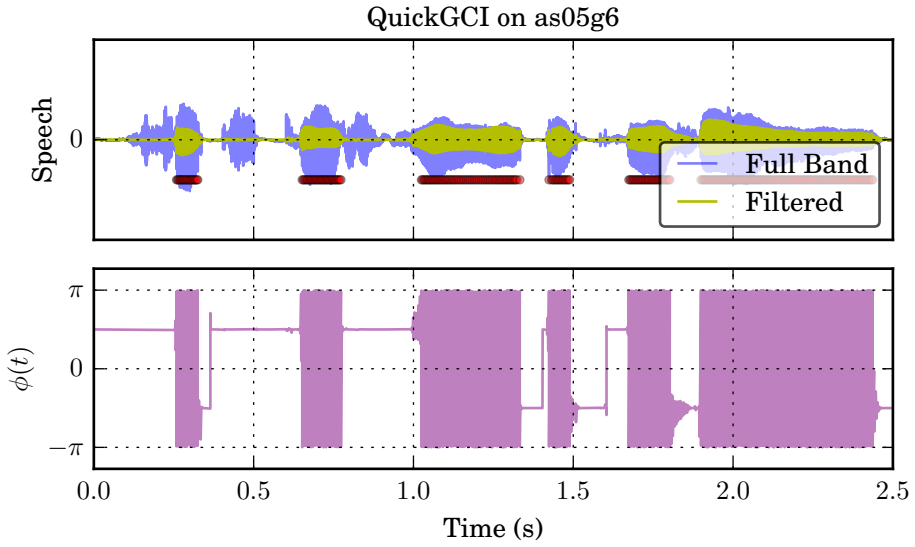


Figure 3.9: A speech sentence ("six plus three equals nine") and its QuickGCI phase angle. GCIs are marked as red circles. Voicing regions have rapid phase accumulation, hence the 2π discontinuities, discernible as solid purple regions at this resolution.

The blue plot shows the full-band speech signal and the yellow overlay shows the bandpass-filtered signal which attenuates the frication portions. The regions of non-voicing (e.g. silence or unvoiced frication) have a phase near $\pm\pi/2$.

[4]Using the peaks in the envelope of the analytic EGG signal was an early iteration of GADFLI, and was used for pre-marking the waveforms database for hand-verification. There were too many cases where the HEGG did not quite work, which warranted further research.

CHAPTER 4

EXPERIMENTAL METHODS

4.1 APLAWD Speech Database

The APLAWD (Archivable Priority List Actual-Word Database) data set [53] provides simultaneous acoustic and electroglottographic recordings of various one-word tokens, two-word tokens, sentences, digits, and letters of the alphabet, spoken by five males and five females.

Its recording format was originally 12-bit at 20 kHz sampling. Its re-released version, APLAWDW [45], provides these waveforms packaged in 16-bit samples in the Microsoft WAV format. Some waveforms are missing, but most are present. There are 10,984 EGG samples available.

Of these 10,984 samples, 40 were discarded from analysis. Five were test tones, 34 had wrap-around and saturation overflow artifacts,[1] and one was just noise (as03d5). A full listing is given in Appendix A.1

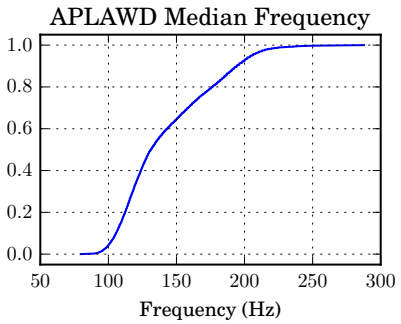
4.1.1 Reference Markings

All 10,944 valid EGG recordings were marked for GCIs. The procedure involved using SIGMA [39] and an early-development version of the proposed algorithm to premark the waveforms, which were then each hand-inspected to remove incorrect GCI markings, add GCI markings when they were missed, and shift markings that were not quite aligned with the GCI. In total, there are 636,665 GCIs in the database. This markings database is publicly available at `https://github.com/serwy/aplawdw`, along with the Python tools used to manipulate and display the data.

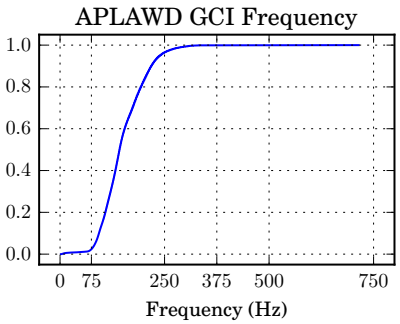
[1]The EGG signals with wrap-around and saturation artifacts can be found by computing the first difference of the EGG signal and finding values with a magnitude greater than 0.5, assuming the 16-bit values are mapped to [-1,1) in floating point. All 34 were manually inspected and determined to have these artifacts.

Figure 4.1(a) shows the sorted median F0 for the EGG samples in the database. These statistics were computed by dividing the sampling rate by the median GCI interval value. Most of the samples have a median F0 between 100-200 Hz.

Figure 4.1(b) shows the instantaneous frequency (sampling rate/GCI interval) for all the samples. This was computed by taking the first difference of the GCI markings for each waveform in the database and aggregating the result. This primitive calculation includes the long gaps between voicing regions (e.g. between words), which shows up as low-frequency values below about 30 Hz. At the other side, fewer than 50 GCI intervals had a frequency greater than 400 Hz.



(a) Cumulative Density Function for the median frequency for all 10,944 (valid) EGG waveforms. Most talkers had F0 between 100-200 Hz.



(b) Cumulative Density Function for the instantaneous frequency computed from consecutive GCI markings. This includes gaps between words (as low-frequency intervals beyond the 600,000 mark). Fewer than 50 GCI intervals were greater than 400 Hz.

Figure 4.1: Statistics of GCIs in the APLAWD EGG corpus from hand-verified GCI markings.

4.1.2 Choice of APLAWD

The APLAWD database has been used for benchmarking the performance of GCI detection algorithms in [38], [39], and [44]. Reusing this database provides the ability to first replicate the results reported by other researchers, and then benchmark the results of newer algorithms.

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 Algorithm Performance Comparison

The performance of GADFLI and QuickGCI is compared to HQTX, TXGEN, SIGMA, and ZFR, using several metrics. The SIGMA algorithm was processed on the original EGG signals and then on 1000 Hz low-pass-filtered[1] EGG signals (SIGMA LPF) to improve its performance. The Hilbert phase method amounts to GADFLI without its thresholding, leaving all the positive-to-negative 2π discontinuities as markers.

There are two ways to quantify hits, misses, and false alarms. One way, herein referred to as "cycle waveform metrics" and used by many authors [42], [5], [48], is to base these metrics on glottal cycles. Each glottal cycle with zero marks is counted as a miss, one mark is counted as a hit, and two or more marks are counted as a single false alarm. Any marks outside of glottal cycles are ignored ([39] mentions a false alarm total metric, which accounts for these spurious markings). Figure 5.1 shows this metric. The sum of the hits, misses, and false alarms equals unity. Table 5.1 shows a comparison of algorithm performance using this cycle metric.

Another way to quantify hits, misses, and false alarms is to consider the entire waveform rather than only the glottal cycles. For each true GCI, the marking closest to it, within a threshold, is counted as a hit. If no markings are within that threshold, then it is a miss. All other markings are false alarms. With this definition, herein referred to as "complete waveform metrics", the sum of hits and misses equals unity. The false alarms are given and expressed as a percentage compared to the number of GCIs in the waveform, and can exceed 100% if there are more markings than GCIs. Table 5.2 shows a comparison using this waveform metric. Everything within

[1]First-order filter, applied forward and backward for zero-phase.

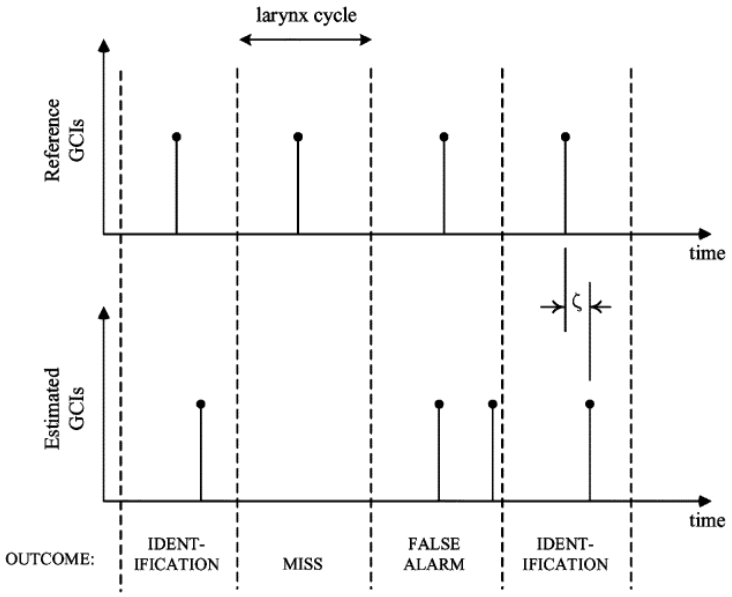


Figure 5.1: Classification criteria from [5], herein referred to as cycle metrics. Identification accurary is measured as ζ.

± 0.5 ms is considered a hit.

All the tested algorithms have a hit rate near or greater than 90%. HQTX and TXGEN trade off miss and alarm rates, where TXGEN has fewer non-cycle markings than HQTX. SIGMA has a large false alarm rate, caused by noise present in the APLAWD recordings. Low-pass-filtering of the EGG before processing (SIGMA LPF) offered better performance by reducing the number of false alarms.

The ZFR method has many of false alarm markings outside of glottal cycle regions. The method, however, has a peak identification rate of near 95%, so that even if all the glottal false alarms were gone, the ZFR performance would still lag SIGMA LPF and HQTX. The ZFR HPF was applied to a 100-Hz HPF DEGG signal, assumed a detrending line of 2 ms, and had its poles at radius $r = 0.999$, to achieve its performance. Not applying the HPF drops its performance to 89% for the cycle hit rate. The drop to 73% for the complete hit rate shows that a large percentage of the markings, while being within the glottal cycle window, were greater than 0.5 ms from the true GCI.

The Hilbert phase method is GADFLI without the inlier elimination and thresholding. It is listed to show the efficacy of using wrapped Hilbert phase discontinuities as the GCI identifying feature and captures the essence of this thesis. The Hilbert phase method, applied to a bandpassed (20-1000 Hz)

Table 5.1: Comparison of EGG GCI algorithm performance with raw numbers and percentages, using the cycle waveform metric for hits, misses, false alarms, and non-glottal markings (other).

Algorithm	Hit	%	Miss	%	False Alarm	%	Other	%
HQTX	623329	97.90	3434	0.54	9902	1.56	40528	6.37
TXGEN	609737	95.77	17597	2.76	9331	1.47	19015	2.99
SIGMA	599631	94.18	2243	0.35	34791	5.46	5840	0.92
SIGMA LPF	630204	98.99	3610	0.57	2817	0.44	10804	1.70
ZFR DEGG	571194	89.72	64369	10.11	1102	0.17	71877	11.29
ZFR HPF	606704	95.29	26087	4.10	3874	0.61	611102	95.98
Hilbert Phase	630477	99.03	25	0.00	6163	0.97	3087755	484.99
GADFLI	634008	99.58	1552	0.24	1105	0.17	4794	0.75
QuickGCI	630982	99.11	4675	0.73	1008	0.16	25208	3.96

EGG signal, has a 99.03% glottal cycle hit rate. Its whole waveform hit rate approaches 99.83%, which suggests that many of the glottal cycle false alarms are low-amplitude and discarded by thresholding. Applying the threshold as done in GADFLI improves glottal cycle performance more than half a percentage point to 99.58% while also greatly reducing the number of spurious markings outside of glottal activity.

QuickGCI, configured for 20-1000 Hz, has a comparable cycle hit rate to GADFLI and to the bare Hilbert phase method. It also has the lowest cycle false alarm rate of all the algorithms. However, its spurious, non-voiced region markings are much higher than GADFLI, comparable to TXGEN, SIGMA LPF, and HQTX.

5.1.1 Timing Errors

The GCI timing errors are shown in the histograms found in Fig. 5.2. These timing errors represent the timing distribution of the hits from Table 5.1.

The HQTX, TXGEN, and QuickGCI algorithms have similar and wider variances when compared to SIGMA and GADFLI.

These timing errors depend on the accuracy of the underlying reference markings. When creating the GCI markings database, SIGMA and a predecessor to GADFLI were applied to each EGG, and their results compared

Table 5.2: Comparison of EGG GCI algorithm performance with raw numbers and percentages, using the complete waveform metric for hits, misses, and false alarms. Markings within ±0.5 ms are considered a hit.

Algorithm	Hit	%	Miss	%	False Alarm	%
HQTX	631407	99.17	5269	0.83	53583	8.42
TXGEN	610355	95.87	26321	4.13	38653	6.07
SIGMA	634386	99.64	2290	0.36	44025	6.91
SIGMA LPF	632935	99.42	3707	0.58	13840	2.17
ZFR DEGG	470702	73.93	165963	26.07	174597	27.42
ZFR HPF	603146	94.74	33520	5.26	622815	97.82
Hilbert Phase	635567	99.83	1098	0.17	3097059	486.45
GADFLI	634604	99.68	2061	0.32	6445	1.01
QuickGCI	625723	98.28	10942	1.72	32509	5.11

and marked on the DEGG waveform. Most of these timing errors span a range of 0.4 ms, which at the 20 kHz sampling rate for APLAWD results in a range of eight samples.

These timing errors also contain within itself variability. SIGMA and GADFLI have the least variability when marking GCIs because the reference markings were pre-populated using these algorithm outputs before being hand-verified and adjusted.

5.1.2 Individual EGG Tokens

The graphs in Fig. 5.3 show the raw error counts for each of the 10944 waveforms. Each curve in a panel is the sorted count of misses, false alarms, or total error (misses+false alarms) for each algorithm. The sorting means that any point along the ordinate may not refer to the same EGG waveform across algorithms.

These panels show that for over half of the APLAWD database, SIGMA, SIGMA LPF, and GADFLI have perfect identification with no spurious markings. Many of the errors come from a smaller subset of EGG signals.

The TXGEN algorithm misses more GCIs than the other algorithms, which behave comparably. The false alarm count differentiates the algorithms. HQTx and QuickGCI have many more spurious markings outside of glottal

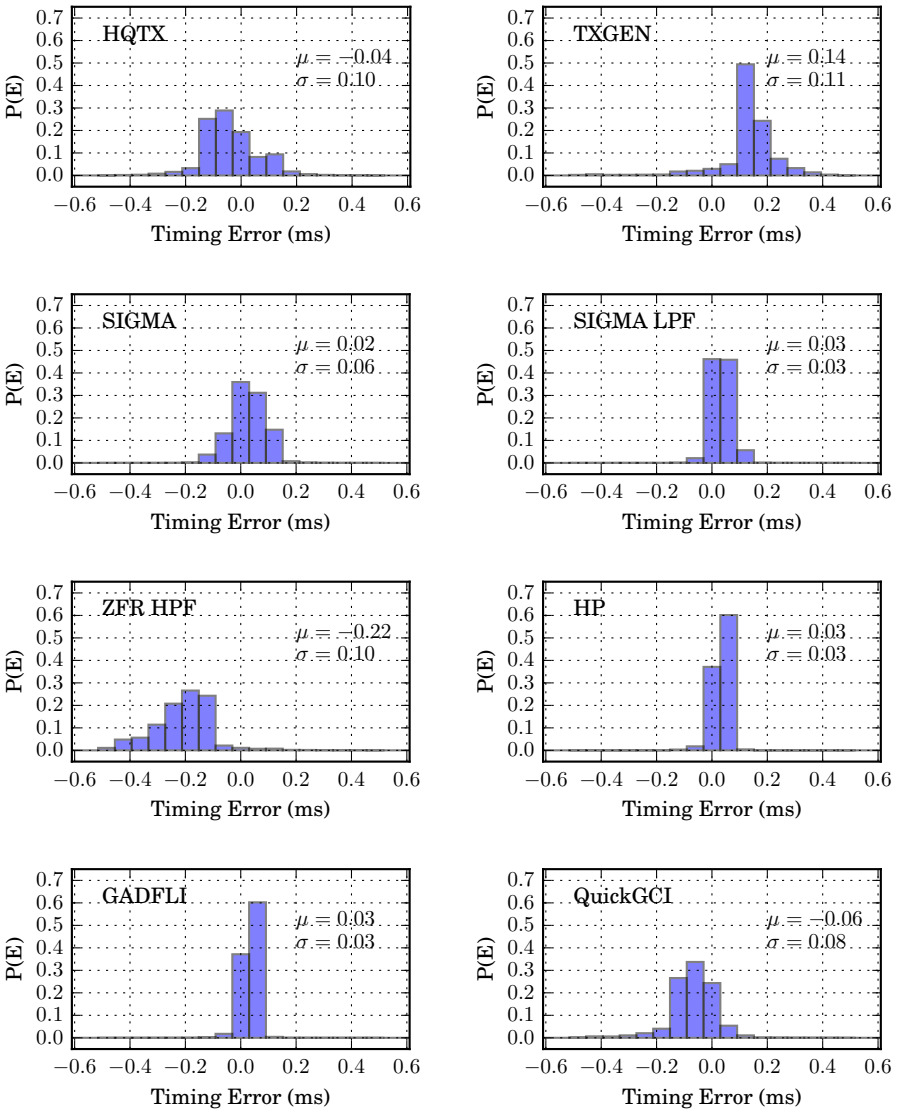


Figure 5.2: Probability distributions of timing errors $P(E)$ for GCI identification across several algorithms. The mean and standard deviation of the error is given in each panel.

cycles. SIGMA has poor false alarm performance for nearly half the samples. Low-pass-filtering the EGG to 1 kHz before applying SIGMA (SIGMA LPF) reduced false alarms, with a negligible increase in misses. The improved performance with filtering suggests that the multi-scale wavelet method is not robust to noise.

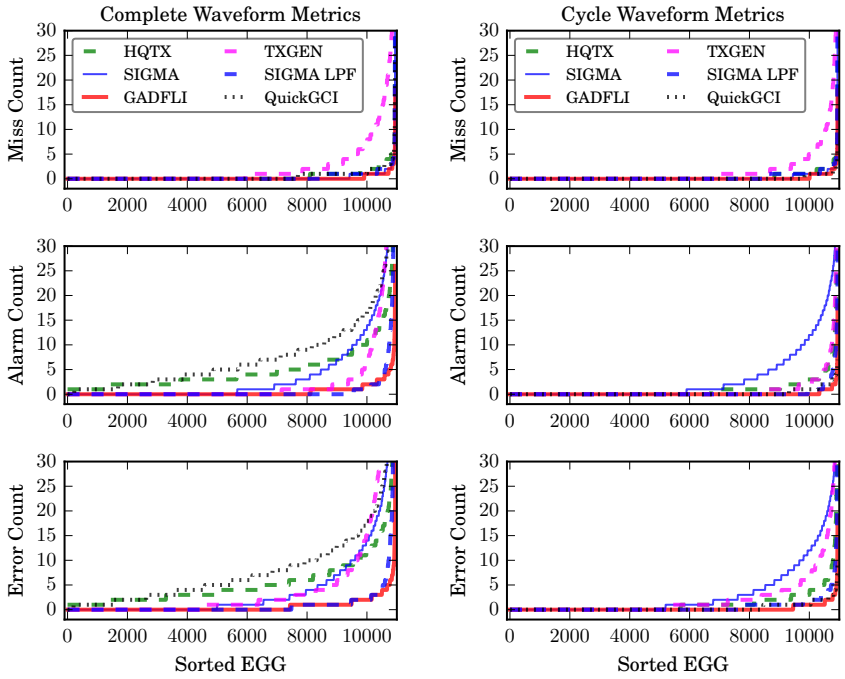


Figure 5.3: Sorted errors for the APLAWD EGG dataset across algorithms. The graphs show the raw count for the particular error type for each of the EGG waveforms, and are sorted along the ordinate. The panels show the sorted miss count, false alarm count, and combined miss and false alarm count (error count).

5.1.3 Varying GADFLI Parameters

The GADFLI algorithm has a few knobs to turn which may or may not seem intuitive. The three main parameters are filter bandwidth, inlier multiplier, and threshold. Figure 5.4 shows how adjusting these parameters affects gottal cycle error metrics. The hit rate (HR) is expressed as 1-HR in order to have a simpler comparison of all the curves. These curves reflect the error as a percentage of the true GCIs in all the APLAWD waveforms.

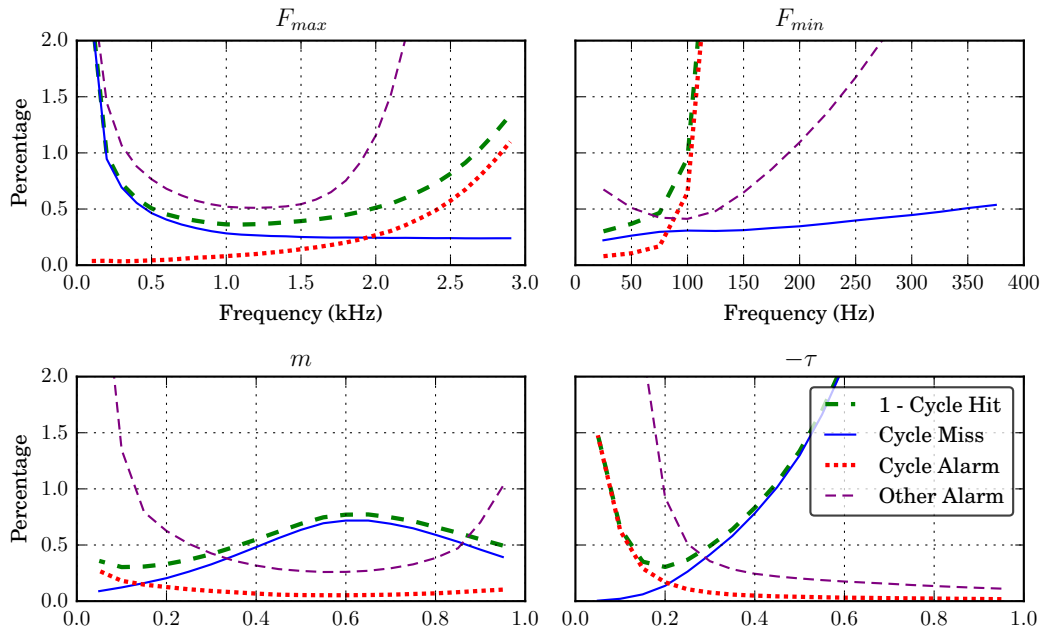


Figure 5.4: GADFLI EGG performance for varying cut-off frequencies F_{max} and F_{min}, amplitude threshold τ, and inlier elimination m. For each graph, the unchanged variables are taken from $F_{min} = 50$ Hz, $F_{max} = 1.25$ kHz, $m = 0.25$, $\tau = -0.25$. All curves are percentages relative to the total number of GCIs.

For frequency, setting the maximum frequency between 1.0-1.5 kHz has a low level of non-cycle markings, which is a useful metric when using GADFLI to generate reference markings for speech performance algorithms. Most of the variability is confined to 99.0-100% over the 0.4-2 kHz range. For the high-pass filter F_{min}, increasing beyond 50 Hz increases the number of cycle false alarms, suggesting that the filtering has over-attenuated the fundamental frequency relative to higher harmonics.

Adjusting the amplitude threshold affects the number of glottal false alarms, hits, and non-cycle false alarms. If non-cycle false alarms are not relevant, the best performance lies at $\tau = -0.2$ for the threshold.

The inlier fraction parameter m may possibly be the most revealing metric. Holding $F_{max} = 1.25$ kHz and $\tau = -0.25$, different values of m have its error metrics constrained between 99.0-100.0% for values between $m = 0.15$ and 0.8. For values of $m > 0.5$, there is much greater variability in the normalization constant as compared to $m < 0.5$.

5.1.4 Varying QuickGCI Parameters

There are two parameters to adjust for QuickGCI, the minimum F_{min} and maximum F_{max} frequencies of the high-pass and low-pass filters. Figure 5.5 shows surface plots of how these two parameters affect QuickGCI cycle waveform metrics. For these surfaces, the measured points were taken from the Cartesian product of $F_{max} \in \{500, 750, 1000, 1500, 2000, 3000\}$ with $F_{min} \in \{20, 50, 75, 100, 150, 200, 300, 400\}$.

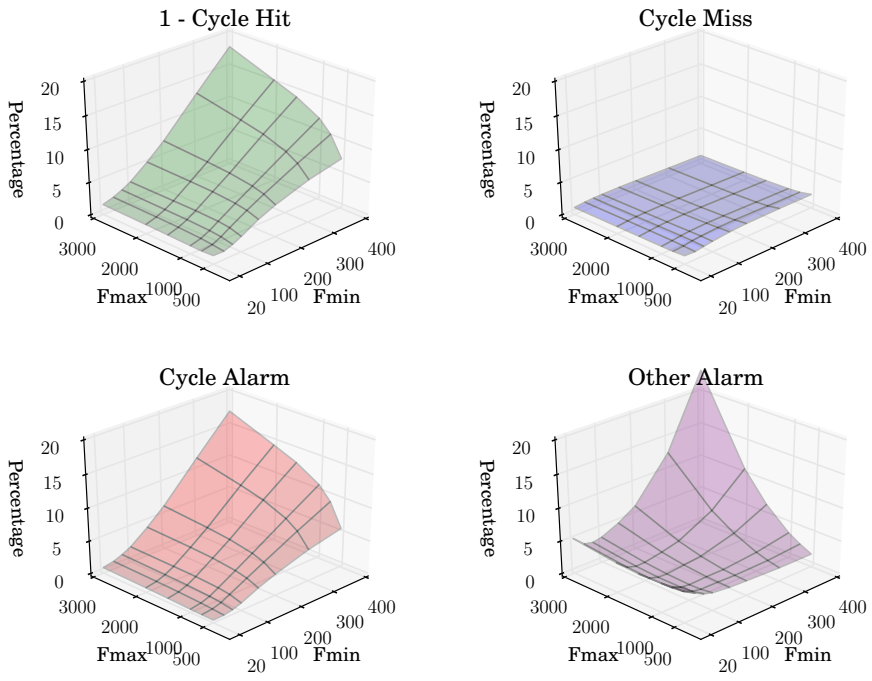


Figure 5.5: Effects of varying QuickGCI bandwidth parameters (in Hz) on cycle metrics, as percentages.

As F_{min} increases, the 1−(cycle hit) and cycle alarm plots follow the same shape, while cycle miss remains flat. This behavior indicates that increasing F_{min} mostly trades hits with alarms. The effects of F_{max} are minimal for low-values of F_{min} and becomes more pronounced with higher F_{min}.

Figure 5.6 shows the behavior of all metrics along a slice, along $F_{min} = 50$ and $F_{max} = 1000$. The cycle miss remains relatively stable around 1%, while the hits and false alarms trade off with the varying filter parameter.

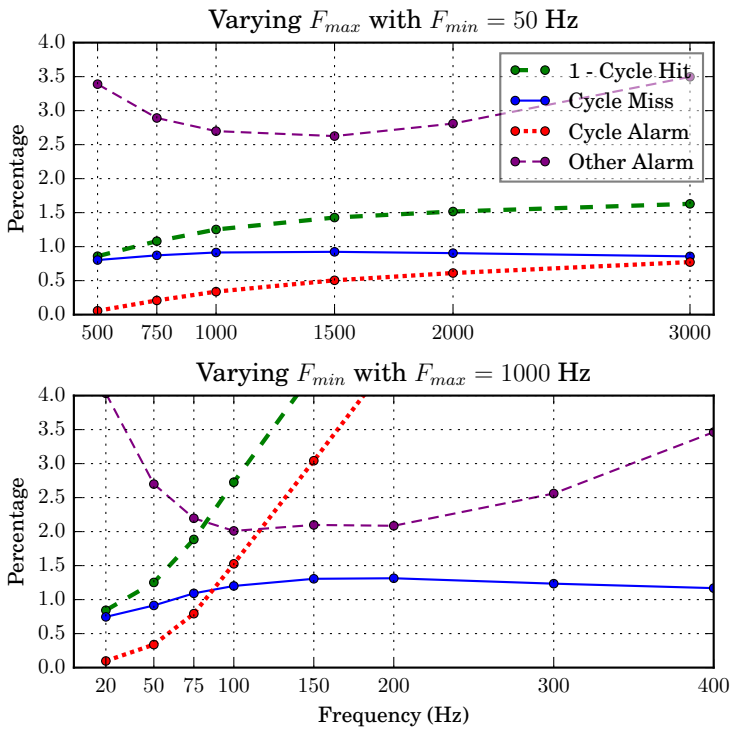


Figure 5.6: QuickGCI EGG performance for varying F_{min} and F_{max} parameters for cycle metrics. The top panel is a slice of the four plots from Fig. 5.5 along $F_{min} = 50$, while the bottom panel slices along $F_{max} = 1000$.

5.1.5 Receiver Operating Characteristics

SIGMA and GADFLI each employ a classification algorithm to separate true GCIs from incorrect markings. SIGMA makes uses of a two-class GMM, taking three feature parameters (linearity of group delay, amplitude, and area) to classify the GCI candidates. GADFLI uses an amplitude thresholding criteria.

The receiver operating characteristic curve [54] can be computed for SIGMA and GADFLI, as shown in Fig. 5.7. Both algorithms were modified to make available the internal candidate selections and classification scoring values. In order to make this analysis tractable, the waveform metrics for hits, misses, and false alarms will be used, as this definition is compatible with ROC curves.

GADFLI has the largest AUC, followed by SIGMA, and then SIGMA LPF, which seemingly contradicts the results in Table 5.2 which shows SIGMA LPF outperforms SIGMA. Directly comparing SIGMA to GADFLI by using

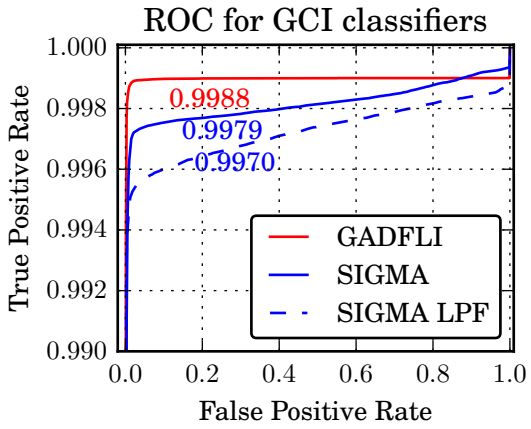


Figure 5.7: Zoomed-in receiver operating characteristic curves for the classifiers used in GADFLI, SIGMA, and SIGMA LPF. The area under the curve (AUC) metrics are given by the curves. The ROC is *not* a good metric for GCI algorithm comparisons due to varying true negative metrics.

the ROCs can be disingenuous, since the candidate selection algorithms are different. The candidate selection algorithms generate the set of data for the classifier stage. Each of these curves in Fig. 5.7 are classifying with different data sets, although these data sets are derived from the same source waveforms. ROC curves are useful for comparing performance of classifiers against the same data set, not different data sets.

The major difference between these GCI candidate data sets is the number of true negatives. Inflating the number of true negatives will necessarily increase the AUC score. Low-pass filtering the EGG signals eliminated many true negatives from consideration, thereby increasing the effect that a false positive has on the FPR curve, which is why SIGMA LPF has a lower AUC than SIGMA, although SIGMA LPF performs better.

True negatives (correct rejections) are not relevant when marking GCIs in a waveform. The hits, misses, and false alarms are the important metrics when evaluating these algorithms.

5.2 Speech Performance

The GADFLI and QuickGCI algorithms were also applied to the speech signals themselves in APLAWD. Speech input signals requires a rotation of

$\theta = -\pi/2$ applied to the analytic signal in order to correctly identify the location of GCIs.[2] The GCIs are found by computing the DEGG's analytic phase angle. For speech, the feature to identify is already present as a negative pulse (see Section 6.4). Rotating by $\theta = -\pi/2$ swaps the real and imaginary components so that the 2π discontinuities align with this negative pulse.

Table 5.3 shows the performance of several speech-oriented GCI detection algorithms. DYPSA, YAGA, and SEDREAMS each make use of the linear prediction residual when computing the location of the GCI.

QuickGCI had its bandpass filter configured from 20-400 Hz for these numbers. GADFLI had its bandpass filter configured from 20-100 Hz, with $m = 0.25$ and $\tau = -0.25$. The ZFR used a fixed detrending line of length 8.7 ms (115 Hz). Two versions of the ZFR were used, the original algorithm (ZFR original), and the same algorithm applied to a high-pass filtered speech signal (ZFR HPF, first-order 100 Hz, zero-phase).

Table 5.3: Comparison of GCI detection from speech signals using several algorithms.

Algorithm	Hit	%	Miss	%	False Alarm	%	Other	%
DYPSA	609279	95.70	8336	1.31	19050	2.99	800430	125.72
YAGA	628585	98.73	1073	0.17	7007	1.10	1045044	164.14
SEDREAMS	628036	98.64	2204	0.35	6425	1.01	412875	64.85
ZFR original	92353	14.51	544142	85.47	170	0.03	28716	4.51
ZFR HPF	629652	98.90	2264	0.36	4749	0.75	243886	38.31
Hilbert Phase	631905	99.25	549	0.09	4211	0.66	363120	57.03
GADFLI	631825	99.24	2099	0.33	2741	0.43	55189	8.67
QuickGCI	625723	98.28	9263	1.45	1679	0.26	42879	6.73

QuickGCI allows for a broader bandwidth as compared to GADFLI and the ZFR due to its rescaling of the signal's envelope which partially attenuates the modulations of the vocal tract relative to the GCI pulse. QuickGCI also has fewer spurious markings in non-glottal-cycle regions, second only to GADFLI.

[2]Both these algorithms were developed for processing EGG signals, and were tried on speech signals as an accident, and almost worked. The rotation term θ was missing.

CHAPTER 6

DISCUSSION

6.1 Expanded SIGMA Analysis

The original analysis for SIGMA in [38] used a hand-marked database of 500 EGG signals chosen from the 10984 available in APLAWDW, and of these hand-marked samples, SIGMA achieved 99.7% accuracy in identification. It is possible that a different subset can result in SIGMA performing terribly, while another subset can have it performing perfectly.

By using the entire APLAWD database, SIGMA was shown to have poor performance for many of the EGG signals, and its root cause was due to high-frequency noise. Preprocessing the EGG signals with a low-pass filter caused SIGMA to have fewer false alarms with a neglible increase in misses.

SIGMA, in its reliance on the GMM for classification, is a non-deterministic algorithm as long as the GMM is initialized randomly. There are EGG samples in APLAWDW where multiple invocations of SIGMA yields drastically different results.

Of the 10944 waveforms, running SIGMA[1] thirty times for each waveform, there were 63 waveforms which had different results, which amounts to 0.57% of the waveforms. Anecdotely, some waveforms required 50 or more invocations before a different results was obtained. This is not a thorough analysis of the possible outputs of SIGMA, since SIGMA has many configuration parameters that affect its output. Figure 6.1 shows an example of how SIGMA can output different results over the 30 invocations.

[1] F_{max} = 400 Hz, F_s = 20000, all other parameters were the defaults found in the `v_sigma.m` file.

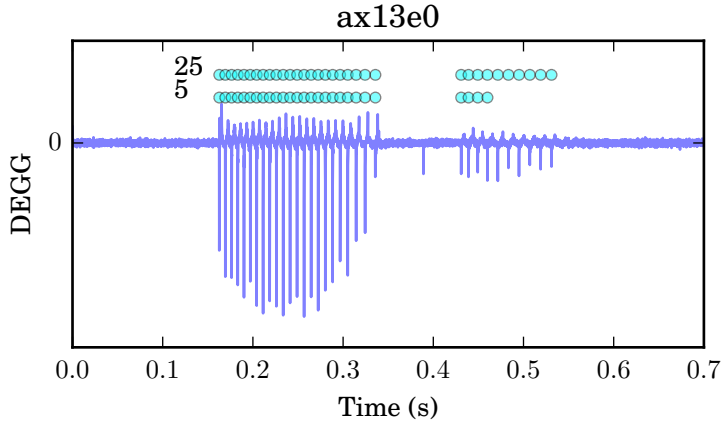


Figure 6.1: SIGMA applied 30 times to ax13e0, with different results. The cyan markings show the two different results, with the top row occurring 25 times, and the bottom row occurring 5 times.

6.2 Zero-Frequency Resonator

The zero-frequency resonator, as specified in [4], was applied to the CMU-Arctic database [55], [43] for its evaluation. Initial attempts at applying the ZFR to APLAWD failed, as the acoustic signals have a lot of low-frequency noise. The original implementation had abysmal performance with a 14.5% hit rate. Preprocessing the speech signal with a 100 Hz, first-order, zero-phase HPF to reduce the effects of this noise improved the ZFR performance to a 98.9% hit rate.

This high-pass filtering is also necessary when processing the CMU-Arctic dataset to achieve comparable performance to reported performance. This crucial filtering step is not mentioned in the ZFR papers, instead attributing all the necessary high-pass filtering to the ZFR's initial first difference computation.

6.3 Hilbert Phase and Zero-Frequency Resonator

The ZFR is a type of Hilbert phase method. The transformation steps for the ZFR, as found in [4], are as follows:

1. Take the first difference of the speech signal:

   $x[n] = s[n] - s[n-1]$

2. Pass the difference speech twice through an ideal resonator:

   $$y_1[n] = -\sum_{k=1}^{2} a_k y_1[n-k] + x[n]$$

   and

   $$y_2[n] = -\sum_{k=1}^{2} a_k y_2[n-k] + y_1[n]$$

   where $a_1 = -2$ and $a_2 = 1$.

3. Remove the trend in the integrator output by removing the local mean of the signal:

   $$y[n] = y_2[n] - \tfrac{1}{2N+1} \sum_{m=-N}^{N} y_2[n+m]$$

The negative-to-positive zero-crossings in the $y[n]$ signal denote the glottal closure instants in the speech signal $s[n]$.

Consider a continuous domain representation of these ZFR steps for a sinusoidal input:

$$s(t) = \cos(\omega t)$$

The first step represents a derivative, giving:

$$x(t) = -\omega \sin(\omega t)$$

The second step represents four successive integrations, giving:

$$y_2(t) = -\sin(\omega t)/\omega^3$$

The third step removes the local mean, and if the local mean interval spans one cycle, it would then evaluate to zero, giving the final equation:

$$y(t) = -\sin(\omega t)/\omega^3$$

The negative-to-positive zero-crossings in $y(t)$ occur every $t = \frac{\pi}{\omega}(2n+1)$, which for the original signal $s(t)$, maps to its minima.

The Hilbert phase approach computes the analytic signal, giving:

$$s_a(t) = \cos(\omega t) + j \sin(\omega t)$$

Its 2π phase discontinuities occur when the imaginary component has a positive-to-negative zero-crossing, which occur every $t = \frac{\pi}{\omega}(2n+1)$, the same as the ZFR. Figure 6.2 shows an example of these signals for $\omega = 1$, with the zero-crossings marked by blue dots.

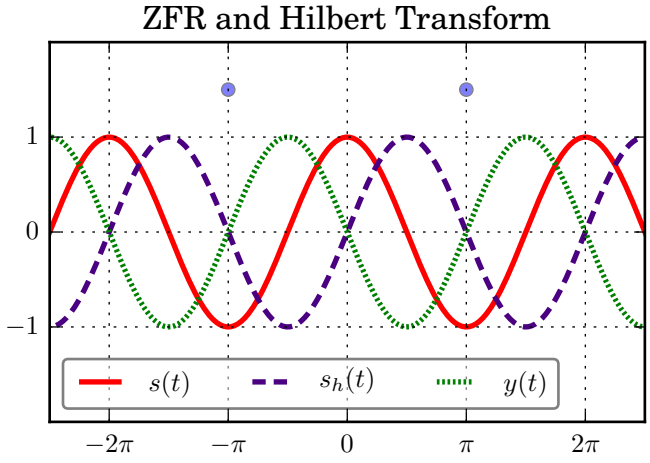


Figure 6.2: Example of ZFR operating on $s(t) = \cos(t)$ and using Hilbert phase discontinuities. Both methods align to the minima of $s(t)$.

Considered broadly, the ZFR, by its transformations, phase-shifts the signal by $\pi/2$, applies a bandpass filter, and tracks zero-crossings. The Hilbert phase method, as used in GADFLI and QuickGCI, performs these same steps, but expressed in the terms of a more general theoretical framework.

The ZFR formulation combines filtering with phase tracking. Its use of zero-Hz resonators creates numerical instability issues, where even the detrending filter cannot return the local oscillations to the axis so that zero-crossings can occur. The original paper [4] used poles with a radius of $r = 0.999$ rather than one to avoid saturating the output of the resonator. A consequence of using the damped pole is the group delay is now spread non-uniformly across frequency rather than being entirely located at $\omega = 0$, causing the detected GCI locations to depend on the fundamental frequency of the speech. The ZFR panel from Fig. 5.2 shows the effects of nonlinear phase filtering delaying the signal by biasing the timing errors.

The ZFR method is not tracking the behavior at 0 Hz, as popularly cited by many papers, but rather phase tracking the fundamental frequency of the speech signal.

The Hilbert phase method splits filtering from its phase tracking and avoids

the numerical instability issues of the ZFR. The Hilbert phase method is the generalization of the ZFR method, and provides a more robust approach for GCI detection.

6.4 Polarity of Speech

Methods of GCI extraction which rely on group delay metrics (see Eq. 2.17) are agnostic to polarity, since the computation of the group delay signal squares the signal, cancelling phase inversion. DYPSA is not sensitive to the polarity of the speech signal. SEDREAMS, YAGA, ZFR, GADFLI, and QuickGCI are sensitive to the polarity of the speech signal.

Let us define positive speech polarity such that a positive excess pressure leaving the lips causes a positive-valued captured acoustic signal. For example, whispering /pa/ with an emphasis on the plosive will create such a pulse. The glottal closure instants are then negative excess pressure pulses. During voicing, the glottis opens and closes rapidly. During the open phase, air flow through the glottis causes a decrease in pressure which then causes the glottis to close. When the glottis closes fully, the upper part of the vocal tract is now isolated from the lungs. The upper vocal tract can be considered a closed volume with an initial pressure condition that is lower than atmospheric pressure. This causes a momentary inflow of volume velocity into the mouth, which causes a negative excess pressure wave to propagate from the lips. This is why the GCI presents itself as a negative pulse.

It is possible to create voicing via inhaling, albeit somewhat strained. The GCIs of an inhaled voice signal are positive.

The ZFR and Hilbert phase methods phase track the negative extrema of the fundamental frequency in the pressure signal, hence their success in identifying GCIs.

CHAPTER 7

CONCLUSIONS

7.1 Conclusion

The wrapped Hilbert phase approach has been proposed as a method for identifying GCIs in EGG signals and speech signals, by means of two algorithms: GADFLI and QuickGCI.

For EGG signals, the performance of the Hilbert phase method by itself (see Table 5.1) shows that the positive-to-negative 2π discontinuities in the analytic phase angle is the feature that denotes the GCI. Coupling this feature with a basic amplitude threshold as done with GADFLI, a reliable set of reference markings from EGG signals can be generated for use with benchmarking speech-only GCI algorithms.

The QuickGCI algorithm provides an alternative approach to identifying GCIs while also rejecting non-speech activity without resorting to an iterative thresholding algorithm. Its performance is comparable, but not quite as good as GADFLI. However, for speech signals, QuickGCI permits a much broader filter bandwidth as compared with GADFLI for comparable performance.

For speech signals, both GADFLI and QuickGCI can identify GCIs with two changes to its parameters: the maximum frequency, and the rotation of the analytic signal (EGG needs no rotation, speech needs $\theta = -\pi/2$) to align the detected Hilbert phase discontinuities to the GCI. The lower maximum frequency de-emphasizes the resonances of the vocal tract which can cause spurious discontinuities in the phase.

This study also introduces an extremely comprehensive database of 10944 reference GCI markings, spanning nearly 160 minutes, for evaluating the performance of GCI detection algorithms for both EGG and speech signals, and is publicly available for inspection and enhancement.

APPENDIX A

ERROR TOKENS IN APLAWD

A.1 Error Tokens

There are 40 erroneous EGG tokens in the APLAWDW [45] database. Five are calibration tones: ac01f0, ac01g0, ac01h0, ac01i0, ac01j0, one is noise: as03d5, and 34 have wrap-around overflow or clipping distortions: ad06e9, ad07a9, al0qa1, al0ta0, al0ta7, as02h6, as02h8, as03a0, as03a9, as04h9, aw05h6, aw06a9, aw14e8, aw19e9, aw23e6, aw42a4, aw42a5, aw42a8, ax03a0, ax07e8, ax07e9, ax13e7, ax14a1, ax14a7, ax14a9, ax25a0, ax28a0, ax32a0, ax32a1, ax32a9, ax35a0, ax35a9, ax38e8, ax43e7.

APPENDIX B

SOURCE CODE

Much of the software research found in this dissertation was completed using the Python 3.5 programming language and its extensive third-party scientific library stack, which includes numpy, pylab, matplotlib, scipy, to name a few. For comparing algorithms, the oct2py package was used to bridge with Python the MATLAB scripts provided by other researchers, such as SIGMA, DYPSA, and SEDREAMS. A more complete listing can be found at `https://github.com/serwy/hilbertgci`. The code here covers the core processing presented in this dissertation.

The full system should be recreate-able using an Intel 80686 virtual machine that can boot Ubuntu 16.04 and install the needed packages that were available at the time of publication (2017). The code contains comments that should allow easily a translation of this code into the standard scientific programming language in use.

The Python code, for the purposes of publication, has made use of tabs so that indentation can be readily inferred. The leading tabs are rendered as lines.

The following code shows the core processing algorithms, as well as the functions used for cycle metrics (`compare_cycle`) and waveform metrics (`compare_markings`). Many of the functions have the named argument `inside`, which is used to toggle the return value to include the locals dictionary, which can be useful for introspecting the internal state transformations of the function.

The default values for keyword arguments were used in the computations presented, unless otherwise stated, e.g. filter frequency boundaries.

File: pyglottal.py

```
from pylab import *
from numpy import *
import scipy.signal as sig

__all__ = ['fasthilbert', 'inlier_elim',
           'butter1', 'gadfli', 'quick_gci',
           'compare_markings', 'compare_cycles',
           'cycle_stats']

def fasthilbert(x):
    # zero-pad to next power of 2...
    L = len(x)
    z = zeros(int(2**(floor(log2(L)) + 1)))
    z[:L] = x
    y = sig.hilbert(z)
    return y[:L]



def inlier_elim(b, m):
    """ Apply inlier elimination, return remaining samples. """
    while True:
        s = b.std()
        idx = find(abs(b) >= m*s)
        if len(idx) != len(b):
            b = b[idx]
        else:
            return b

def butter1(fc, btype='low'):
    """ Generate a 1st order Butterworth filter. """
    # equivalent to sig.butter(1, fc, btype)
    a1 = -(1 - tan(pi*fc/2)) / (1 + tan(pi*fc/2))
    A = array([1, a1])
    if btype == 'low':
```

```
    bl = (1+a1) / 2
    B = array([bl, bl])
  elif btype == 'high':
    bh = (1-a1) / 2
    B = array([bh, -bh])
  return B, A


def der(x):
  """ Apply a first difference. """
  dx = sig.lfilter([1, -1], [1], x)
  dx[0] = dx[1] # avoid initial spike
  return dx


def gadfli(g, fmin=20, fmax=1000, fs=20000, m=0.25, tau=-0.25,
           theta=0, reps=2, inside=False):
  """ Return GCIs using GADFLI algorithm. """

  Bh, Ah = butter1(fmin/(fs/2), 'high')
  Bl, Al = butter1(fmax/(fs/2), 'low')

  for i in range(reps):
    g = sig.filtfilt(Bh, Ah, g)
    g = sig.filtfilt(Bl, Al, g)

  dg = der(g)
  h = fasthilbert(dg) * exp(1j * theta)

  dphi = der(angle(h))
  gci_c = find(dphi < -1.5*pi) # candidates

  rh = real(h)
  kept = inlier_elim(rh, m)
  scale = kept.std()
```

```
    gci = array([i for i in gci_c if rh[i] < tau*scale])

    if not inside: return gci
    else: return gci, locals()


def quick_gci(g, fmin=20, fmax=1000, fs=20000, theta=0,
              reps=2, reps2=None, inside=False):
    """ Return GCIs using QuickGCI algorithm. """

    Bh, Ah = butter1(fmin/(fs/2), 'high')
    Bl, Al = butter1(fmax/(fs/2), 'low')

    for i in range(reps):
        g = sig.filtfilt(Bh, Ah, g)
        g = sig.filtfilt(Bl, Al, g)

    x = fasthilbert(g) * exp(1j*theta)
    q = abs(x) * imag(-x)

    for i in range(reps if reps2 is None else reps2):
        q = sig.filtfilt(Bl, Al, q)

    r = fasthilbert(q)

    dphi = der(angle(r))
    gci = find(dphi < -1.5*pi)

    if not inside: return gci
    else: return gci, locals()


def _get_bounds(x, y, idx, half=True):
    """ return search boundaries for x[idx]"""
    mid = x[idx]
    if idx == 0:
```

```
        lo = min([x[0], y[0]]) - 1
    else:
        lo = x[idx-1]
        if half:
            lo = lo + (mid - lo) // 2
    if idx == len(x) - 1:
        hi = max([x[-1], y[-1]]) + 1
    else:
        hi = x[idx+1]
        if half:
            hi = mid + (hi - mid) // 2
    return lo, hi


def _get_match(x, y, half=True):
    """ return matches between x and y """
    match = []
    for x_idx, a in enumerate(x):
        lo_x, hi_x = _get_bounds(x, y, x_idx, half)
        y_win = [n for n, v in enumerate(y)
                 if lo_x < v <= hi_x]
        for y_idx in y_win:
            lo_y, hi_y = _get_bounds(y, x, y_idx, half)
            b = y[y_idx]
            if lo_y < a <= hi_y:
                match.append((a, b))
    return match

def compare_markings(x, y, thresh=None, inside=False):
    match_half = _get_match(x, y, half=True)
    match_full = _get_match(x, y, half=False)
    match_conflict = set(match_full) - set(match_half)

    keep = match_half.copy()
    for lb, ub in match_conflict:
        for i,j in keep:
```

```
            if i==lb or j==ub:
                break
        else:
            #both end points not already matched
            keep.append((lb, ub))

    if thresh: # purge matches too distant
        keep = [(i,j) for i,j in keep if abs(i-j) <= thresh]

    cx, cy = zip(*keep) if keep else ([], [])
    x_only = sorted(set(x) - set(cx))
    y_only = sorted(set(y) - set(cy))

    cd = [(i, (j-i)) for i,j in keep]
    common, diff = zip(*cd) if cd else ([], [])
    r = tuple(map(array, [x_only, y_only, common, diff]))

    if not inside: return r
    else: return r, locals()

def compare_cycles(x, y, HP, vt=True, centered=False):
    """ Find all markings in y within a glottal cycle
        derived from GCI markings in x, to within a
        maximum half-period in samples.
    """

    x = sorted(x)
    y = sorted(y)

    cycles = {i:[] for i in x}
    bounds = {}
    other = []

    # handle edge cases in x
    if not x: return {}, {}, y
    x = [x[0] - 3*HP] + x + [x[-1] + 3*HP]
```

```
    b = 0

    for a in range(1, len(x)-1):
        lo, mid, hi = x[a-1], x[a], x[a+1]
        # Adjust bounds to be halfway between
        # previous and next GCI markings.
        lo = lo + (mid - lo) // 2
        hi = mid + (hi - mid) // 2

        if mid - HP > lo:
            if vt: # voicing transition
                # likely at onset of voicing,
                # use next period
                lo = max([mid-HP, 2*mid-hi])
            else:
                lo = mid - HP

        if mid + HP < hi:
            if vt:
                # likely at offset of voicing,
                # use previous period
                hi = min([mid+HP, 2*mid-lo])
            else:
                hi = mid + HP

        if centered:
            # shrink bounds such that mid is centered
            d = min([hi-mid, mid-lo])
            lo, hi = mid-d, mid+d

        bounds[mid] = (lo, hi)
        # find all the indices in y within bounds
        while b < len(y) and y[b] < hi:
            if y[b] >= lo:
                cycles[mid].append(y[b])
            else:
```

```
        other.append(y[b])
        b += 1

    other.extend(y[b:])  # leftover
    return cycles, bounds, other

def cycle_stats(cycles):
    """ Compute the number of hits, misses, alarms
        using the output of compare_cycles."""
    v = list(map(len, cycles.values()))
    hit   = v.count(1)
    miss  = v.count(0)
    alarm = len(v) - miss - hit
    return hit, miss, alarm
```

REFERENCES

[1] J. Yadav and K. S. Rao, "Detection of vowel offset point from speech signal," *IEEE Signal Processing Letters*, vol. 20, no. 4, pp. 299–302, April 2013.

[2] H. Valbret, E. Moulines, and J. P. Tubach, "Voice transformation using psola technique," in *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Mar 1992, pp. 145–148.

[3] D. G. Silva, L. C. Oliveira, and M. Andrea, "Jitter estimation algorithms for detection of pathological voices," *EURASIP J. Adv. Signal Process*, vol. 2009, pp. 9:1–9:9, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1155/2009/567875

[4] K. S. R. Murty and B. Yegnanarayana, "Epoch extraction from speech signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1602–1613, Nov 2008.

[5] P. A. Naylor, A. Kounoudes, J. Gudnason, and M. Brookes, "Estimation of glottal closure instants in voiced speech using the DYPSA algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 34–43, Jan 2007.

[6] R. Sharma, R. K., and S. R. M. Prasanna, "Analysis of electroglottograph signal using ensemble empirical mode decomposition," in *2014 Annual IEEE India Conference (INDICON)*, Dec 2014, pp. 1–6.

[7] S. Hahn, *Hilbert Transforms in Signal Processing*, ser. Artech House signal processing library. Artech House, 1996. [Online]. Available: https://books.google.com/books?id=b_RSAAAAMAAJ

[8] D. Lauria and C. Pisani, "On Hilbert transform methods for low frequency oscillations detection," *IET Generation, Transmission Distribution*, vol. 8, no. 6, pp. 1061–1074, June 2014.

[9] J. Tribolet, "A new phase unwrapping algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 2, pp. 170–177, Apr 1977.

[10] B. Bhanu and J. McClellan, "On the computation of the complex cepstrum," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 5, pp. 583–585, Oct 1980.

[11] K. Steiglitz and B. Dickinson, "Phase unwrapping by factorization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, no. 6, pp. 984–991, Dec 1982.

[12] M. Alavi-Sereshki and J. Prabhakar, "A tabulation of Hilbert transforms for electrical engineers," *IEEE Transactions on Communications*, vol. 20, no. 6, pp. 1194–1198, Dec 1972.

[13] J. Proakis and D. Manolakis, *Digital Signal Processing, 4th ed.* Prentice Hall, 2007.

[14] S. C. Kak, "The discrete Hilbert transform," *Proceedings of the IEEE*, vol. 58, no. 4, pp. 585–586, April 1970.

[15] V. Cizek, "Discrete Hilbert transform," *IEEE Transactions on Audio and Electroacoustics*, vol. 18, no. 4, pp. 340–343, Dec 1970.

[16] S. L. Marple, "Computing the discrete-time 'analytic' signal via FFT," in *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers (Cat. No.97CB36136)*, vol. 2, Nov 1997, pp. 1322–1325.

[17] M. Elfataoui and G. Mirchandani, "A frequency-domain method for generation of discrete-time analytic signals," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3343–3352, Sept 2006.

[18] B. Gold, A. V. Oppenheim, and C. M. Rader, *Theory and Implementation of the Discrete Hilbert Transform.* MIT Press, 1969, pp. 14–42. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6301522

[19] J. Proakis and M. Salehi, *Communication Systems Engineering*, ser. Pearson Education. Prentice Hall, 2002. [Online]. Available: https://books.google.com/books?id=8WqfQgAACAAJ

[20] P. Fabre, "Un procédé électrique percutané d'inscription de l'accolement glottique au cours de la phonation: glottographie de haute fréquence," *Bulletin de l'Académie Nationale de Médecine*, vol. 141, pp. 66–69, 1957.

[21] P. Kitzing, "Clinical applications of electroglottography," *Journal of Voice*, vol. 4, no. 3, pp. 238–249, 1990.

[22] A. M. Smith and D. G. Childers, "Laryngeal evaluation using features from speech and the electroglottograph," *IEEE Transactions on Biomedical Engineering*, vol. BME-30, no. 11, pp. 755–759, Nov 1983.

[23] A. Askenfelt, J. Gauffin, P. Kitzing, and J. Sundberg, "Electroglottograph and contact microphone for measuring vocal pitch," *Speech Transmission Laboratory, Quarterly Progress and Status Report*, vol. 4, pp. 13–21, 1977.

[24] M. Rothenberg, "A multichannel electroglottograph," *Journal of Voice*, vol. 6, no. 1, pp. 36 – 43, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0892199705800074

[25] D. G. Childers, D. M. Hicks, G. P. Moore, and Y. A. Alsaka, "A model for vocal fold vibratory motion, contact area, and the electroglottogram," *The Journal of the Acoustical Society of America*, vol. 80, no. 5, pp. 1309–1320, 1986. [Online]. Available: http://scitation.aip.org/content/asa/journal/jasa/80/5/10.1121/1.394382

[26] D. Childers, D. Hicks, G. Moore, L. Eskenazi, and A. Lalwani, "Electroglottography and vocal fold physiology," *Journal of Speech, Language, and Hearing Research*, vol. 33, no. 2, pp. 245–254, 1990.

[27] R. H. Colton and E. G. Conture, "Problems and pitfalls of electroglottography," *Journal of Voice*, vol. 4, no. 1, pp. 10 – 24, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0892199705800773

[28] D. G. Childers and J. N. Larar, "Electroglottography for laryngeal function assessment and speech analysis," *IEEE Transactions on Biomedical Engineering*, vol. BME-31, no. 12, pp. 807–817, Dec 1984.

[29] N. Henrich, C. d'Alessandro, B. Doval, and M. Castellengo, "On the use of the derivative of electroglottographic signals for characterization of nonpathological phonation," *The Journal of the Acoustical Society of America*, vol. 115, no. 3, pp. 1321–1332, 2004. [Online]. Available: http://scitation.aip.org/content/asa/journal/jasa/115/3/10.1121/1.1646401

[30] C. T. Herbst, H. Herzel, J. G. Švec, M. T. Wyman, and W. T. Fitch, "Visualization of system dynamics using phasegrams," *Journal of The Royal Society Interface*, vol. 10, no. 85, 2013. [Online]. Available: http://rsif.royalsocietypublishing.org/content/10/85/20130288

[31] C. T. Herbst, W. T. S. Fitch, and J. G. Švec, "Electroglottographic wavegrams: A technique for visualizing vocal fold dynamics noninvasively)," *The Journal of the Acoustical Society of America*, vol. 128, no. 5, pp. 3070–3078, 2010. [Online]. Available: http://scitation.aip.org/content/asa/journal/jasa/128/5/10.1121/1.3493423

[32] A. Krishnamurthy and D. Childers, "Two-channel speech analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 730–743, 1986.

[33] K. E. Barner, "Nonlinear estimation of degg signals with applications to speech pitch detection," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 4, Oct 1996, pp. 2243–2246.

[34] C. Mooshammer, "Acoustic and laryngographic measures of the laryngeal reflexes of linguistic prominence and vocal effort in germana)," *The Journal of the Acoustical Society of America*, vol. 127, no. 2, pp. 1047–1058, 2010. [Online]. Available: http://scitation.aip.org/content/asa/journal/jasa/127/2/10.1121/1.3277160

[35] L. Rabiner, "On the use of autocorrelation analysis for pitch detection," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 1, pp. 24–33, Feb 1977.

[36] L. Rabiner, M. Cheng, A. Rosenberg, and C. McGonegal, "A comparative performance study of several pitch detection algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 5, pp. 399–418, Oct 1976.

[37] W. Hess and H. Indefrey, "Accurate time-domain pitch determination of speech signals by means of a laryngograph," *Speech Communication*, vol. 6, no. 1, pp. 55 – 68, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0167639387900690

[38] M. R. P. Thomas and P. A. Naylor, "The SIGMA algorithm for estimation of reference-quality glottal closure instants from electroglottograph signals," in *Signal Processing Conference, 2008 16th European*, Aug 2008, pp. 1–5.

[39] M. R. P. Thomas and P. A. Naylor, "The SIGMA algorithm: A glottal activity detector for electroglottographic signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 8, pp. 1557–1566, Nov 2009.

[40] M. Brookes, P. A. Naylor, and J. Gudnason, "A quantitative assessment of group delay methods for identifying glottal closures in voiced speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 456–466, March 2006.

[41] M. A. Huckvale, "Speech filing system: Tools for speech," http://www.phon.ucl.ac.uk/resource/sfs/, 1987.

[42] A. Kounoudes, P. A. Naylor, and M. Brookes, "The DYPSA algorithm for estimation of glottal closure instants in voiced speech," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 1, May 2002, pp. I–349–I–352.

[43] "CMU-Arctic speech synthesis databases," http://festvox.org/cmu arctic/index.html, 2003.

[44] T. B. Amin and P. Marziliano, "Glottal activity detection using finite rate of innovation methods," in *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on*, Dec 2013, pp. 1–5.

[45] M. Brookes, "APLAWDW (archivable priority list actual-word database in wav format)," http://www.commsp.ee.ic.ac.uk/ sap/resources/aplawdw/, 2015.

[46] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds1," *Speech Communication*, vol. 27, no. 3–4, pp. 187 – 207, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639398000855

[47] K. S. S. Srinivas and K. Prahallad, "An FIR implementation of zero frequency filtering of speech signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2613–2617, Nov 2012.

[48] M. R. P. Thomas, J. Gudnason, and P. A. Naylor, "Estimation of glottal closing and opening instants in voiced speech using the yaga algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 82–91, Jan 2012.

[49] T. Drugman and T. Dutoit, "Glottal closure and opening instant detection from speech signals," in *Interspeech*, 2009, pp. 2891–2894.

[50] T. Drugman, M. Thomas, J. Gudnason, P. Naylor, and T. Dutoit, "Detection of glottal closure instants from speech signals: A quantitative review," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 994–1006, March 2012.

[51] I. Jtc, "Sc22/wg14. iso/iec 9899: 2011," *Information technology—Programming languages—C. http://www. iso. org/iso/iso_catalogue/catalogue_ tc/catalogue_detail. htm*, 2011.

[52] D. Govind, S. R. M. Prasanna, and K. Ramesh, "Improved method for epoch extraction in high pass filtered speech," in *2013 Annual IEEE India Conference (INDICON)*, Dec 2013, pp. 1–5.

[53] G. Lindsey, A. Breen, and S. Nevard, "Spar's archivable actual-word databases," *University College London," Technical Report*, 1987.

[54] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861 – 874, 2006, ROC Analysis in Pattern Recognition. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016786550500303X

[55] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.